IEC 62061

Edition 2.1 2024-03
CONSOLIDATED VERSION

INTERNATIONAL STANDARD

colour inside

Safety of machinery – Functional safety of safety-related control systems

IEC 62061:2021-03+AMD1:2024-03 CSV(en)

IEC Secretariat
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee, …). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch
Discover our powerful search engine and read freely all the publications previews, graphical symbols and the glossary. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 500 terminological entries in English and French, with equivalent terms in 25 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC 62061

Edition 2.1 2024-03
CONSOLIDATED VERSION

INTERNATIONAL STANDARD

colour inside

Safety of machinery – Functional safety of safety-related control systems

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 13.110; 25.040.99; 29.020

ISBN 978-2-8322-8675-3

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

SAFETY OF MACHINERY – FUNCTIONAL SAFETY OF SAFETY-RELATED CONTROL SYSTEMS

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) IEC draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). IEC takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, IEC had not received notice of (a) patent(s), which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at https://patents.iec.ch. IEC shall not be held responsible for identifying any or all such patent rights.

This consolidated version of the official IEC Standard and its amendment has been prepared for user convenience.

IEC 62061 edition 2.1 contains the second edition (2021-03) [documents 44/885/FDIS and 44/888/RVD] and its amendment 1 (2024-03) [documents 44/1020/FDIS and 44/1024/RVD].

In this Redline version, a vertical line in the margin shows where the technical content is modified by amendment 1. Additions are in green text, deletions are in strikethrough red text. A separate Final version with all changes accepted is available in this publication.

IEC 62061 has been prepared by IEC technical committee 44: Safety of machinery – Electrotechnical aspects. It is an International Standard.

This second constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

– structure has been changed and contents have been updated to reflect the design process of the safety function,
– standard extended to non-electrical technologies,
– definitions updated to be aligned with IEC 61508-4,
– functional safety plan introduced and configuration management updated (Clause 4),
– requirements on parametrization expanded (Clause 6),
– reference to requirements on security added (Subclause 6.8),
– requirements on periodic testing added (Subclause 6.9),
– various improvements and clarification on architectures and reliability calculations (Clause 6 and Clause 7),
– shift from "SILCL" to "maximum SIL" of a subsystem (Clause 7),
– use cases for software described including requirements (Clause 8),
– requirements on independence for software verification (Clause 8) and validation activities (Clause 9) added,
– new informative annex with examples (Annex G),
– new informative annexes on typical $MTTF_D$ values, diagnostics and calculation methods for the architectures (Annex C, Annex D and Annex H).

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/standardsdev/publications.

The committee has decided that the contents of this document and its amendment will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn, or
- revised.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

As a result of automation, demand for increased production and reduced operator physical effort, Safety-related Control Systems (referred to as SCS) of machines play an increasing role in the achievement of overall machine safety. Furthermore, the SCS themselves increasingly employ complex electronic technology.

IEC 62061 specifies requirements for the design and implementation of safety-related control systems of machinery. This document is machine sector specific within the framework of IEC 61508.

NOTE While IEC 62061 and ISO 13849-1 are using different methodologies for the design of safety related control systems, they intend to achieve the same risk reduction.

This International Standard is intended for use by machinery designers, control system manufacturers and integrators, and others involved in the specification, design and validation of an SCS. It sets out an approach and provides requirements to achieve the necessary performance and facilitates the specification of the safety functions intended to achieve the risk reduction.

This document provides a machine sector specific framework for functional safety of an SCS of machines. It only covers those aspects of the safety lifecycle that are related to safety requirements allocation through to safety validation. Requirements are provided for information for safe use of SCS of machines that can also be relevant to later phases of the lifecycle of an SCS.

There are many situations on machines where SCS are employed as part of safety measures that have been provided to achieve risk reduction. A typical case is the use of an interlocking guard that, when it is opened to allow access to the danger zone, signals the safety related parts of the machine control system to stop hazardous machine operation. In automation, the machine control system that is used to achieve correct operation of the machine process often contributes to safety by mitigating risks associated with hazards arising directly from control system failures. This document gives a methodology and requirements to:

- assign the required safety integrity for each safety function to be implemented by SCS;
- enable the design of the SCS appropriate to the assigned safety (control) function(s);
- integrate safety-related subsystems designed in accordance with other applicable functional safety-related standards (see 6.3.4);
- validate the SCS.

This document is intended to be used within the framework of systematic risk reduction, in conjunction with risk assessment described in ISO 12100. Suggested methodologies for a safety integrity assignment are given in informative Annex A.

SAFETY OF MACHINERY – FUNCTIONAL SAFETY OF SAFETY-RELATED CONTROL SYSTEMS

1 Scope

This International Standard specifies requirements and makes recommendations for the design, integration and validation of safety-related control systems (SCS) for machines. It is applicable to control systems used, either singly or in combination, to carry out safety functions on machines that are not portable by hand while working, including a group of machines working together in a co-ordinated manner.

This document is a machinery sector specific standard within the framework of IEC 61508 (all parts).

The design of complex programmable electronic subsystems or subsystem elements is not within the scope of this document. This is in the scope of IEC 61508 or standards linked to it; see Figure 1.

NOTE 1 Elements such as systems on chip or microcontroller boards are considered complex programmable electronic subsystems.

The main body of this sector standard specifies general requirements for the design, and verification of a safety-related control system intended to be used in high/continuous demand mode.

This document:

– is concerned only with functional safety requirements intended to reduce the risk of hazardous situations;

– is restricted to risks arising directly from the hazards of the machine itself or from a group of machines working together in a co-ordinated manner;

NOTE 2 Requirements to mitigate risks arising from other hazards are provided in relevant sector standards. For example, where a machine(s) is part of a process activity, additional information is available in IEC 61511.

This document does not cover

– electrical hazards arising from the electrical control equipment itself (e.g. electric shock – see IEC 60204-1);

– other safety requirements necessary at the machine level such as safeguarding;

– specific measures for security aspects – see IEC ~~TR~~ TS 63074.

This document is not intended to limit or inhibit technological advancement.

Figure 1 illustrates the scope of this document.

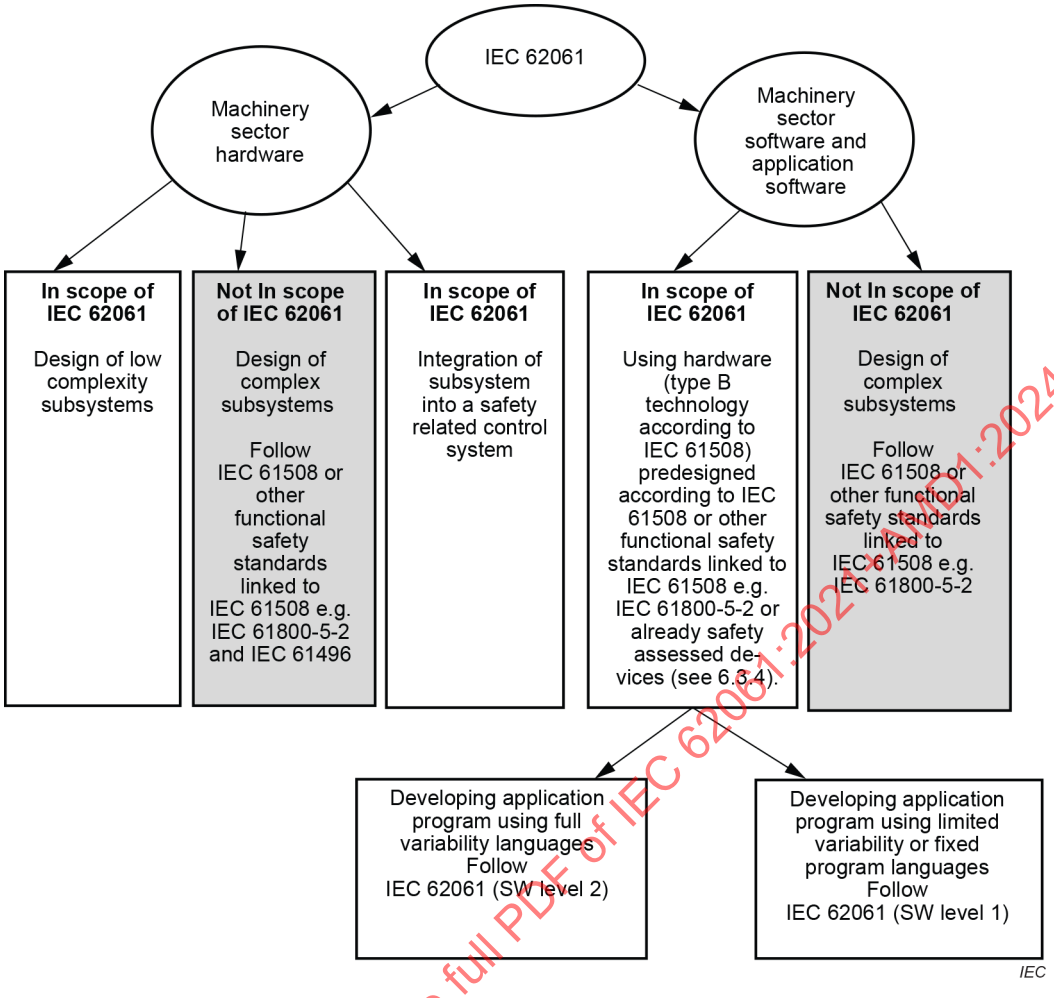


Figure 1 – Scope of this document

2 Normative references

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

IEC 60204-1:2016, *Safety of machinery – Electrical equipment of machines – Part 1: General requirements*

IEC 61000-1-2:2016, *Electromagnetic compatibility (EMC) – Part 1-2: General – Methodology for the achievement of functional safety of electrical and electronic systems including equipment with regard to electromagnetic phenomena*

IEC 61508 (all parts), *Functional safety of electrical/electronic/programmable electronic safety-related systems*

IEC 61508-2:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems*

IEC 61508-3:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements*

ISO 12100:2010, *Safety of machinery – General principles for design – Risk assessment and risk reduction*

ISO 13849 (all parts), *Safety of machinery – Safety-related parts of control systems*

ISO 13849-1:2015, *Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design*

ISO 13849-2:2012, *Safety of machinery – Safety-related parts of control systems – Part 2: Validation*

3 Terms, definitions and abbreviations

3.1 Alphabetical list of definitions

Terms used throughout IEC 62061 are given in Table 1. Also included are some common abbreviations related to machinery safety.

Table 1 – Terms used in IEC 62061

Term	Definition number
application software	3.2.59
architectural constraint	3.2.46
architecture	3.2.45
average frequency of dangerous failure per hour (PFH)	3.2.29
average probability of dangerous failure on demand (PFD_{avg})	3.2.31
baseline (configuration)	3.2.67
bypass	3.2.17
common cause failure (CCF)	3.2.56
complex component	3.2.8
configuration management	3.2.66
continuous mode	3.2.28
dangerous failure	3.2.52
demand	3.2.25
diagnostic coverage (DC)	3.2.49
diagnostic test interval	3.2.50
embedded software	3.2.60
failure	3.2.51
fault	3.2.33
fault tolerance	3.2.34
full variability language (FVL)	3.2.61
functional safety	3.2.10
hardware fault tolerance (HFT)	3.2.35
hardware safety integrity	3.2.22
harm	3.2.12
hazard	3.2.11
high demand mode	3.2.27
integrator	3.2.13

Term	Definition number
limited variability language (LVL)	3.2.62
low complexity component	3.2.7
low demand mode	3.2.26
machine control system	3.2.2
machinery (machine)	3.2.1
mean repair time (MRT)	3.2.40
mean time to failure ($MTTF$)	3.2.37
mean time to dangerous failure ($MTTF_D$)	3.2.38
mean time to restoration (MTTR)	3.2.39
muting	3.2.16
pre-designed (SCS or subsystem)	3.2.5
probability of dangerous failure on demand (PFD)	3.2.30
process safety time	3.2.41
proof test coverage	3.2.48
proof test	3.2.47
protective measure	3.2.14
random hardware failure	3.2.57
ratio of dangerous failure (RDF)	3.2.55
risk	3.2.15
safe failure	3.2.53
safe failure fraction (SFF)	3.2.54
safe state	3.2.68
safety	3.2.9
safety function	3.2.18
safety integrity	3.2.21
safety integrity level (SIL)	3.2.24
safety-related control system (SCS)	3.2.3
safety-related software	3.2.63
security	3.2.69
(SCS) diagnostic function	3.2.19
(SCS) fault reaction function	3.2.20
subsystem	3.2.4
subsystem element	3.2.6
sub-function	3.2.36
systematic failure	3.2.58
systematic safety integrity	3.2.23
target failure measure	3.2.32
useful lifetime	3.2.42
validation (of the safety function)	3.2.65
verification	3.2.64
well-tried component	3.2.43
well-tried safety principles	3.2.44

3.2 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

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

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

3.2.1
machinery
machine
assembly, fitted with or intended to be fitted with a drive system consisting of linked parts or components, at least one of which moves, and which are joined together for a specific application

Note 1 to entry: The term “machinery” also covers an assembly of machines which, in order to achieve the same end, are arranged and controlled so that they function as an integral whole.

[SOURCE: ISO 12100:2010, 3.1]

3.2.2
machine control system
system that responds to input signals from the machinery and/or from an operator and generates output signals causing the machinery to operate in the desired manner

Note 1 to entry: The machine control system includes input devices and final elements.

[SOURCE: IEC 61508-4:2010, 3.3.3, modified – the term defined has been changed, "process" has been changed to "machinery"]

3.2.3
safety-related control system
SCS
part of the control system of a machine which implements a safety function by one or more subsystems

Note 1 to entry: SCS is similar to SRECS of the previous edition of this document.

3.2.4
subsystem
entity of the top-level architectural design of a safety-related system where a dangerous failure of the subsystem results in dangerous failure of a safety function

Note 1 to entry: This differs from common language where “subsystem” may mean any sub-divided part of an entity, the term “subsystem” is used in this document within a strongly defined hierarchy of terminology: “subsystem” is the first level subdivision of a system. The parts resulting from further subdivision of a subsystem are called “subsystem elements”.

Note 2 to entry: A complete subsystem can be made up from a number of identifiable and separate subsystem elements.

Note 3 to entry: The subsystem specification includes its role in the safety function and its interface with the other subsystems of the SCS.

Note 4 to entry: One subsystem can be part of several safety functions, e.g. the same combination of contactors can be used to de-energise a motor either in the event of detection of a person in a danger zone or also in the event of opening an interlock guard.

[SOURCE: IEC 61508-4:2010, 3.4.4, modified – cross references removed and notes added]

3.2.5
pre-designed SCS or subsystem
SCS or subsystem which meets the relevant requirements of a functional safety standard

3.2.6
subsystem element
part of a subsystem, comprising a single component or any group of components

Note 1 to entry: A subsystem element may comprise hardware and software.

Note 2 to entry: Elements that are not directly necessary for the safety function are not included, but may support it (for example, filters elements, protection against over-voltage).

Note 3 to entry: A subsystem element is the lowest level of detail to consider when ensuring that the requirements of a sub-function are met.

3.2.7
low complexity component/subsystem
component/subsystem in which

– the failure modes are well-defined; and
– the behaviour under fault conditions can be completely defined

Note 1 to entry: Behaviour of the low complexity component / subsystem under fault conditions may be determined by analytical and/or test methods.

Note 2 to entry: Examples of low complexity components / subsystem are limit switches, electro-mechanical relays or contactors.

[SOURCE: IEC 61508-4:2010, 3.4.3, modified – the term defined has been changed, leading to reformulation of text. Example converted into note 2]

3.2.8
complex component / subsystem
component / subsystem in which

– the failure modes are not well-defined; or
– the behaviour under fault conditions cannot be completely defined

3.2.9
safety
freedom from unacceptable risk

[SOURCE: IEC 61508-4:2010, 3.1.11]

3.2.10
functional safety
part of the overall safety of the machine and the machine control system that depends on the correct functioning of the SCS and other risk reduction measures

[SOURCE: IEC 61508-4:2010, 3.1.12, modified – using terms machine, machine control system and SCS]

3.2.11
hazard
potential source of harm

Note 1 to entry: The term hazard can be qualified in order to define its origin or the nature of the expected harm (e.g. electric shock hazard, crushing hazard, cutting hazard, toxic hazard, fire hazard).

[SOURCE: ISO 12100:2010, 3.6, modified – note 1 has been modified and notes 2 and 3 deleted]

3.2.12
harm
injury or damage to the health of people

[SOURCE: ISO/IEC Guide 51:2014, 3.1, modified – without damage to property or the environment]

3.2.13
integrator
entity who designs, manufactures or assembles an integrated manufacturing system and is responsible for the safety strategy, including the protective measures, control interfaces and interconnections of the control system

Note 1 to entry: The integrator may be for example a manufacturer, assembler, engineering company, or entity with the overall responsibility for the machine.

[SOURCE: ISO 11161:2007, 3.10, modified – "provides" has been deleted, last entry in note reformulated]

3.2.14
protective measure
measure intended to achieve risk reduction

[SOURCE: ISO 12100:2010, 3.19, modified – bullet list removed]

3.2.15
risk
combination of the probability of occurrence of harm and the severity of that harm

[SOURCE: ISO/IEC Guide 51:2014, 3.9 modified – note to entry removed]

3.2.16
muting
temporary automatic suspension of a safety function(s)

Note 1 to entry: Other means are used to maintain the safety level.

[SOURCE: ISO 13849-1:2015, 3.1.8, modified – "by the SRP/CS" has been deleted, note added]

3.2.17
bypass
action or facility to prevent all or parts of the SCS functionality from being executed

Note 1 to entry: Examples of bypassing include:

– the input signal is blocked from the trip logic while still presenting the input parameters and alarm to the operator;
– the output signal from the trip logic to a final element is held in the normal state preventing final element operation;
– a physical bypass line is provided around the final element;
– preselected input state (e.g., on/off input) or set is forced by means of an engineering tool (e.g., in the application program).

Note 2 to entry: Other terms are also used to refer to bypassing, such as override, defeat, disable, force, or inhibit or muting.

[SOURCE: IEC 61511-1:2016, 3.2.4, modified – SIS replaced by SCS]

3.2.18
safety function
function implemented by an SCS with a specified integrity level that is intended to maintain the safe condition of the machine or prevent an immediate increase of the risk(s) in respect of a specific hazardous event

Note 1 to entry: This term is used instead of "safety-related control function (SRCF)" of IEC 62061:2015. This definition differs from ISO 12100 because this document addresses risk reduction performed by SCS.

Note 2 to entry: A safety function is typically starting with a detection and evaluation of an 'initiation event' and ending with an output causing a reaction of a 'machine actuator'.

Note 3 to entry: Parts of machine operating function(s), e.g. the reaction of a machine actuator, can also be part of safety function(s).

[SOURCE: IEC 61508-4:2010, 3.5.1, modified – terminology adapted to machinery, other risk reduction measures deleted, example deleted, notes added]

3.2.19
(SCS) diagnostic function
function intended to detect faults in the SCS and initiate a specified fault reaction function when a fault is detected

Note 1 to entry: This function is intended to detect faults that could lead to a dangerous failure of a safety function and initiate a specified fault reaction function.

3.2.20
(SCS) fault reaction function
function that is initiated when a fault within an SCS is detected by the SCS diagnostic function

3.2.21
safety integrity
probability of an SCS or its subsystem satisfactorily performing the required safety function under all stated conditions within a stated period of time

Note 1 to entry: The higher the level of safety integrity of the item, the lower the probability that the item will fail to carry out the required safety function.

Note 2 to entry: Safety integrity comprises hardware safety integrity and systematic safety integrity.

[SOURCE: IEC 61508-4:2010, 3.5.4, modified – terminology adapted to machinery, notes 2, 3, 5 deleted]

3.2.22
hardware safety integrity
part of the safety integrity of an SCS or its subsystems relating to random hardware failures in a dangerous mode of failure

Note 1 to entry: The term relates to failures in a dangerous mode, that is, those failures of a safety-related system that would impair its safety integrity.

Note 2 to entry: Hardware safety integrity includes architectural constraints.

[SOURCE: IEC 61508-4:2010, 3.5.7 – terminology adapted to machinery, note 1 shortened, note 2 added]

3.2.23
systematic safety integrity
part of the safety integrity of an SCS or its subsystems relating to its resistance to systematic failures in a dangerous mode

Note 1 to entry: Systematic safety integrity cannot usually be quantified precisely.

Note 2 to entry: Requirements for systematic safety integrity apply to both hardware and software aspects of an SCS or its subsystems.

[SOURCE: IEC 61508-4:2010, 3.5.6, modified – terminology adapted to machinery, note 1 shortened, note 2 added]

3.2.24
safety integrity level
SIL
discrete level (one out of a possible three) for describing the capability to perform a safety function where safety integrity level three has the highest level of safety integrity and safety integrity level one has the lowest

3.2.25
demand
event that causes the SCS to perform a safety function

Note 1 to entry: Demand mode means that a safety function is only performed on request (demand) in order to transfer the machine into a specified state. The SCS does not influence the machine until there is a demand on the safety function.

Note 2 to entry: Demand rate (DR) or the frequency of demands is one of the main factor that is considered for assessing the demand mode, low or high. For this particular purpose, the demand rate (DR) can be identified with the rate of events, where harm would occur without intervention of the safety function. This rate may be lower than an actual rate of triggering the safety function during operation.

Note 3 to entry: For an emergency stop function, the demand mode is not defined. To determine the achieved SIL, the principle for evaluation of the selected demand mode of the other functions is usually applicable.

3.2.26
low demand mode
mode of operation in which the frequency of demands of a safety function is no greater than one per year

[SOURCE: IEC 61508-4:2010, 3.5.16, modified – low demand extracted from definition of "mode of operation"]

3.2.27
high demand mode
mode of operation in which the frequency of demands of a safety function is greater than one per year

Note 1 to entry: Continuous mode means that a safety function is performed continuously, i.e. the SCS is continuously controlling the machine and a (dangerous) failure of its function can result in a hazard.

Note 2 to entry: The distinction between high demand and continuous mode is relevant for the qualification of diagnostic measures (refer to 7.4.3 and 7.4.4). It is not relevant for target failure measure and SIL assignment.

[SOURCE: IEC 61508-4:2010, 3.5.16, modified – high demand extracted from definition of "mode of operation", notes added]

3.2.28
continuous mode
mode of operation where the safety function retains the machinery in a safe state as a part of normal operation

Note 1 to entry: Continuous mode means that a safety function is performed continuously, i.e. the SCS is continuously controlling the machine and a (dangerous) failure of its function can result in a hazard.

Note 2 to entry: The distinction between high demand and continuous mode is relevant for the qualification of diagnostic measures (refer to 7.4.3 and 7.4.4). It is not relevant for target failure measure and SIL assignment".

[SOURCE: IEC 61508-4:2010, 3.5.16, modified – high demand extracted from definition of "mode of operation", notes added]

3.2.29
average frequency of a dangerous failure per hour
***PFH* or PFH_D**
average frequency of dangerous failure of an SCS to perform a specified safety function over a given period of time

Note 1 to entry: Both terms *PFH* and PFH_D correspond to the probability of dangerous failures per hour (IEC 62061:2005+AMD1:2012+AMD2:2015).

Note 2 to entry: The term “average probability of dangerous failure per hour” is not used in this edition anymore but the acronym *PFH* has been retained but when it is used it means “average frequency of dangerous failure [h]".

[SOURCE: IEC 61508-4:2010, 3.6.19, modified – terminology adapted to machinery, existing notes deleted, new notes added]

3.2.30
probability of dangerous failure on demand
PFD
safety unavailability (see IEC 60050-192) of an SCS to perform the specified safety function when a demand occurs from the machinery or machinery control system

Note 1 to entry: The [instantaneous] unavailability (as per IEC 60050-192) is the probability that an item is not in a state to perform a required function under given conditions at a given instant of time, assuming that the required external resources are provided. It is generally noted by U (t).

Note 2 to entry: The [instantaneous] availability does not depend on the states (running or failed) experienced by the item before it. It characterizes an item which only has to be able to work when it is required to do so, for example, an SCS working in low demand mode.

Note 3 to entry: If periodically tested, the *PFD* of an SCS is, in respect of the specified safety function, represented by a saw tooth curve with a large range of probabilities ranging from low, just after a test, to a maximum just before a test.

[SOURCE: IEC 61508-4:2010, 3.6.17, modified – terminology adapted to machinery]

3.2.31
average probability of dangerous failure on demand
PFD_{avg}
mean unavailability (see IEC 60050-192) of an SCS to perform the specified safety function when a demand occurs from the machinery or machinery control system as an average over time

Note 1 to entry: The mean unavailability over a given time interval [t1, t2] is generally noted by U (t1, t2).

Note 2 to entry: Two kind of failures contribute to *PFD* and PFD_{avg}: the dangerous undetected failures occurred since the last proof test and genuine on demand failures caused by the demands (proof tests and safety demands) themselves. The first one is time dependent and characterized by their dangerous failure rate λ_{DU}(t) whilst the second one is dependent only on the number of demands and is characterized by a probability of failure per demand (denoted by γ).

Note 3 to entry: As genuine on demand failures cannot be detected by tests, it is necessary to identify them and take them into consideration when calculating the target failure measures.

[SOURCE: IEC 61508-4:2010, 3.6.18, modified – terminology adapted to machinery]

3.2.32
target failure measure
intended *PFH* or PFD_{avg} to be achieved to meet a specific safety integrity requirement(s)

Note 1 to entry: Target failure measure is specified in terms of:

– the average probability of a dangerous failure of the safety function on demand, (for a low demand mode of operation);

– the average frequency of a dangerous failure [h^{-1}] (for a high demand mode of operation or a continuous mode of operation).

[SOURCE: IEC 61508-4:2010, 3.5.17, modified – "target probability of dangerous mode failures" changed to "intended PFH or PFD_{avg}", bullet list moved to note 1, existing note deleted]

3.2.33
fault
abnormal condition that may cause a reduction in, or loss of, the capability of an SCS, a subsystem, or a subsystem element to perform a required function

Note 1 to entry: In IEC 60050-192, 192-04-01 a fault of an item is described as inability to perform as required, due to an internal state.

[SOURCE: IEC 61508-4:2010, 3.6.1, modified – terminology adapted to machinery, note shortened]

3.2.34
fault tolerance
ability of an SCS, a subsystem, or subsystem element to continue to perform a required function in the presence of faults or failures

[SOURCE: IEC 61508-4:2010, 3.6.3, modified – terminology adapted to machinery]

3.2.35
hardware fault tolerance
HFT
property of a subsystem to potentially lose the safety function upon at least N+1 faults

Note 1 to entry: A hardware fault tolerance of N means that N+1 faults of a subsystem could cause a loss of the safety function.

[SOURCE: IEC 61508-2:2010, derived from 7.4.4.1.1]

3.2.36
sub-function
part of a safety function whose failure can result in a failure of the safety function

Note 1 to entry: In this document, a safety function can be seen as a logical AND of the sub-functions.

3.2.37
mean time to failure
MTTF
expectation of the mean time to failure

Note 1 to entry: $MTTF$ is normally expressed as an average value of expectation of the time to failure.

[SOURCE: IEC 60050-192, 192-05-11, modified – note added and original notes removed]

3.2.38
mean time to dangerous failure
***MTTF*$_D$**
expectation of the mean time to dangerous failure

[SOURCE: Definition derived from IEC 60050-192, 192-05-11, modified – restricted to dangerous failures]

3.2.39
mean time to restoration
MTTR
expected time to achieve restoration after a fault has occurred in a safety function.

Note 1 to entry: MTTR encompasses:

- the time to detect the failure (a); and
- the time spent before starting the repair (b); and
- the effective time to repair (c); and
- the time before the component is put back into operation (d).

The start time for (b) is the end of (a); the start time for (c) is the end of (b); the start time for (d) is the end of (c).

Note 2 to entry: During this time the machine can continue to operate

[SOURCE: IEC 61508-4:2010, 3.6.21, modified – terminology adapted to machinery and more details added to definition]

3.2.40
mean repair time
MRT
mean repair time after a fault has been detected in a safety function and machine continues to operate

Note 1 to entry: MRT encompasses:

- the time spent before starting the repair (b); and
- the effective time to repair (c); and
- the time before the component is put back into operation (d).

Note 2 to entry: Depending on the type of detected fault and the fault reaction, the numerical values for MRT and MTTR can be different.

[SOURCE: IEC 61508-4:2010, 3.6.22, modified – terminology adapted to machinery and more details added to definition, note 1 made similar to 3.2.39, note 2 added]

3.2.41
process safety time
period of time between a failure, that has the potential to give rise to a hazardous event, occurring in the machinery or machinery control system and the time by which action has to be completed in the machinery to prevent the hazardous event occurring

Note 1 to entry: It is foreseen that the safety function detects the failure and completes its action soon enough to prevent the hazardous event taking into account any process lag (e.g. stopping times).

[SOURCE: IEC 61508-4:2010, 3.6.20 modified – terminology adapted to machinery, note 1 added]

3.2.42
useful lifetime
minimum elapsed time between the installation of the SCS or subsystem or subsystem element and the point in time when component failure rates of the SCS or subsystem or subsystem element can no longer be predicted, with any accuracy

Note 1 to entry: Typically it will be 20 years or less unless the manufacturers of the SCS and its subsystems can justify a longer lifetime by providing evidence, based on calculations, showing that reliability data is valid for the longer lifetime.

[SOURCE: IEC 61131-6:2012, 3.57, modified – terminology adapted to machinery, note 1 added, example deleted]

3.2.43
well-tried component
for a safety-related application, component for a safety-related application which has been either

a) widely used in the past with successful results in similar safety-related applications as given as well-tried components in the informative annexes of ISO 13849-2, or

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

Note 1 to entry: ISO 13849-2 lists a variety of components and the conditions for specific technologies under which the component can be considered well-tried.

Note 2 to entry: Newly developed components may be considered as equivalent to “well-tried” if they fulfil the conditions of b).

Note 3 to entry: The decision to accept a particular component as being “well-tried” depends on the application, e.g. owing to the environmental influences and can be impacted by product or manufacturer changes.

Note 4 to entry: Complex electronic components (e.g. PLC, microprocessor, application-specific integrated circuit) cannot be considered as equivalent to “well tried”.

Note 5 to entry: A well-tried component is not a proven in use component.

3.2.44
well-tried safety principles
principles that have proved effective in the design or integration of safety-related control systems in the past, to avoid or control critical faults or failures which can influence the performance of a safety function

Note 1 to entry: Newly developed safety principles can be considered as equivalent to “well-tried” if they are verified using principles which demonstrate their suitability and reliability for safety-related applications.

Note 2 to entry: Well-tried safety principles are effective not only against random hardware failures, but also against systematic failures which may creep into the product at some point in the course of the product life cycle, e.g. faults arising during product design, integration, modification or deterioration.

Note 3 to entry: Tables A.2, B.2, C.2 and D.2 in the informative annexes of ISO 13849-2:2012 address well-tried safety principles for different technologies.

[SOURCE: Definition derived from ISO 13849-1:2015]

3.2.45
architecture
specific configuration of hardware and software elements in an SCS

[SOURCE: IEC 61508-4:2010, 3.3.4, modified – terminology adapted to machinery]

3.2.46
architectural constraint
set of architectural requirements that limit the SIL that can be claimed for a subsystem

3.2.47
proof test
periodic test that can detect dangerous undetected faults and degradation in an SCS and its subsystems so that, if necessary, the relevant parts of the SCS and its subsystems can be restored to an “as new” condition or as close as practical to this condition

Note 1 to entry: A proof test is intended to confirm that relevant parts of an SCS are in a condition that assures the specified safety integrity.

Note 2 to entry: The effectiveness of the proof test will be dependent both on failure coverage and repair effectiveness. In practice, detecting 100 % of the degradation that could lead to the hidden dangerous failures later on is not easily achieved. For complex elements or safety features that are difficult to verify, a proof test coverage of 100 % cannot be usually obtained.

[SOURCE: IEC 61508-4:2010, 3.8.5, modified – terminology adapted to machinery, notes 1, 3, 4 deleted, new note 1 added, note 2 shortened]

3.2.48
proof test coverage
term given to the percentage of dangerous undetected failures that are detected by a defined proof test procedure

Note 1 to entry: It measures the effectiveness of a proof test and ranges from 0 % to 100 % (perfect proof-test).

Note 2 to entry: For example, a PTC of 95 % states that 95 % of all possible undetected failures will be detected during the proof test. It doesn't include aging or degradation not directly related to the safety function failure.

Note 3 to entry: The PTC can be estimated by the means of Failure Mode and Effects Analysis (FMEA) in conjunction with engineering judgement based on sound evidence.

3.2.49
diagnostic coverage
DC
fraction of dangerous failures detected by automatic on-line diagnostic tests

Note 1 to entry: The fraction of dangerous failures is computed by using the dangerous failure rates associated with the detected dangerous failures divided by the total rate of dangerous failures.

Note 2 to entry: The dangerous failure diagnostic coverage is computed using the following equation, where DC is the diagnostic coverage, λ_{DD} is the detected dangerous failure rate and λ_{Dtotal} is the total dangerous failure rate:

$$DC = \frac{\sum \lambda_{DD}}{\sum \lambda_{Dtotal}} \quad (1)$$

Note 3 to entry: This definition is applicable providing the individual components have constant failure rates.

[SOURCE: IEC 61508-4:2010, 3.8.6, modified – part of the definition has been moved to a note to entry]

3.2.50
diagnostic test interval
interval between on-line tests to detect faults in a subsystem that has a specified diagnostic coverage

[SOURCE: IEC 61508-4:2010, 3.8.7, modified – replacing safety-related system by subsystem]

3.2.51
failure
termination of the ability of an item (SCS, a subsystem or a subsystem element) to perform a required function

Note 1 to entry: Failures are either random (in hardware) or systematic (in hardware or software).

Note 2 to entry: After a failure, the item has a fault.

Note 3 to entry: "Failure" is an event, as distinguished from "fault", which is a state.

Note 4 to entry: The concept of failure as defined does not apply to items consisting of software only.

[SOURCE: IEC 61508-4:2010, 3.6.4, modified and ISO 12100-1:2010, 3.32]

3.2.52
dangerous failure
failure of an SCS, a subsystem, or a subsystem element that plays a part in implementing the safety function that:

a) prevents a safety function from operating when required (demand mode) or causes a safety function to fail (continuous mode) such that the machine is put into a hazardous or potentially hazardous state; or

b) decreases the probability that the safety function operates correctly when required

[SOURCE: IEC 61508-4:2010, 3.6.47, modified – Terminology adapted to machinery ~~and figure replaced by textual description and ISO 12100-1:2010, 3.34~~]

3.2.53
safe failure
failure of an SCS, a subsystem, or a subsystem element that plays a part in implementing the safety function that:

a) results in the spurious operation of the safety function to put the machine (or part thereof) into a safe state or maintain a safe state; or

b) increases the probability of the spurious operation of the safety function to put the machine (or part thereof) into a safe state or maintain a safe state

[SOURCE: IEC 61508-4:2010, 3.6.8, modified – terminology adapted to machinery]

3.2.54
safe failure fraction
SFF
fraction of the overall failure rate of a subsystem that does not result in a dangerous failure

Note 1 to entry: The diagnostic coverage (if any) of each subsystem in SCS is taken into account in the calculation of the probability of random hardware failures. The safe failure fraction is taken into account when determining the architectural constraints on hardware safety integrity (see 7.4).

Note 2 to entry: "No effect failures" and "no part failures" (see IEC 61508-4) is not used for SFF calculations.

3.2.55
ratio of dangerous failure
RDF
fraction of the overall failure rate of an element that can result in a dangerous failure

3.2.56
common cause failure
CCF
failure, that is the result of one or more events, causing concurrent failures of two or more separate channels in a multiple channel subsystem, leading to failure of a safety function

[SOURCE: IEC 61508-4:2010, 3.6.10, modified – system failure replaced by failure of a safety function]

3.2.57
random hardware failure
failure occurring at a random time, which results from one or more of the possible degradation mechanisms in the hardware

[SOURCE: IEC 61508-4:2010, 3.6.5, modified – notes removed]

3.2.58
systematic failure
failure, related in a deterministic way to a certain cause, which can only be eliminated by a modification of the design or of the manufacturing process, operational procedures, documentation or other relevant factors

Note 1 to entry: Corrective maintenance without modification will usually not eliminate the failure cause.

Note 2 to entry: A systematic failure can be induced by simulating the failure cause.

Note 3 to entry: Examples of causes of systematic failures include human error in

- the safety requirements specification;
- the design, manufacture, installation and/or operation of the hardware;
- the design and/or implementation of the software.

[SOURCE: IEC 61508-4:2010, 3.6.6, modified – note 3 slightly changed, note 4 removed]

3.2.59
application software
software specific to the application, that is implemented by the designer of the SCS, generally containing logic sequences, limits and expressions that control the appropriate input, output, calculations, and decisions necessary to meet the SCS functional requirements

3.2.60
embedded software
software, supplied as part of a pre-designed subsystem, that is not intended to be modified and that relates to the functioning of, and services provided by, the SCS or subsystem, as opposed to the application software

Note 1 to entry: Firmware and system software are examples of embedded software.

3.2.61
full variability language
FVL
type of language that provides the capability to implement a wide variety of functions and applications

Note 1 to entry: Typical example of systems using FVL are general-purpose computers.

Note 2 to entry: FVL is normally found in embedded software and is rarely used in application software.

Note 3 to entry: FVL examples include: Ada, C, Pascal, Instruction List, assembler languages, C++, Java, SQL.

[SOURCE: IEC 61511-1:2016, 3.2.75.3, modified – first part of definition supressed and link to process sector deleted]

3.2.62
limited variability language
LVL
type of language that provides the capability to combine predefined, application specific, library functions to implement the safety requirements specifications

Note 1 to entry: A LVL provides a close functional correspondence with the functions required to achieve the application.

Note 2 to entry: Typical examples of LVL are given in IEC 61131-3. They include ladder diagram, function block diagram and sequential function chart. Instruction lists and structured text are not considered to be LVL.

Note 3 to entry: Typical example of systems using LVL: Programmable Logic Controller (PLC) configured for machine control.

[SOURCE: IEC 61511-1:2016, 3.2.75.2, modified – note 1 turned into definition, note 2 deleted, note 3 replaced]

3.2.63
safety-related software
software that is used to implement safety functions in a safety-related system

3.2.64
verification
confirmation by examination (e.g. tests, analysis) that the SCS, its subsystems or subsystem elements meet the requirements set by the relevant specification

EXAMPLE: Verification activities include

- reviews on outputs (documents from all phases) to ensure compliance with the objectives and requirements of the phase, taking into account the specific inputs to that phase;
- design reviews;
- tests performed on the designed products to ensure that they perform according to their specification;
- integration tests performed where different parts of a system are put together in a step-by-step manner and by the performance of environmental tests to ensure that all the parts work together in the specified manner.

[SOURCE: IEC 61508-4:2010, 3.8.1, modified – terminology adapted to machinery, note deleted]

3.2.65
validation (of the safety function)
confirmation by examination (e.g. tests, analysis) that the SCS meets the functional safety requirements of the specific application

[SOURCE: IEC 61508-4:2010, 3.8.2, modified – terminology adapted to machinery, notes deleted]

3.2.66
configuration management
discipline of identifying the components of an evolving system for the purposes of controlling changes to those components and maintaining continuity and traceability throughout the lifecycle

[SOURCE: IEC 61508-4:2010, 3.7.3, modified – note removed]

3.2.67
baseline (configuration)
well-defined set of elements (hardware, software, documentation, tests, etc.) of an SCS at a specific point in time.

Note 1 to entry: A baseline serves as a basis for verification, validation, modification and changes.

Note 2 to entry: If an element is changed, the status of the baseline is intermediate until a new baseline is defined.

3.2.68
safe state
state of the machine when safety is achieved

Note 1 to entry: The safe state doesn't include the restoration of initial equipment failures.

[SOURCE: IEC 61508-4:2010, 3.1.13, modified – terminology adapted to machinery, original note deleted, note 1 added]

3.2.69
security
1) measures taken to protect a system
2) condition of a system that results from the establishment and maintenance of measures to protect the system
3) condition of system resources being free from unauthorized access and from unauthorized or accidental change, destruction, or loss

4) capability of a computer-based system to provide adequate confidence that unauthorized persons and systems can neither modify the software and its data nor gain access to the system functions, and yet to ensure that this is not denied to authorized persons and systems

5) prevention of illegal or unwanted penetration of, or interference with the proper and intended operation of an industrial automation and control system

Note 1 to entry: Measures can be controls related to physical security (controlling physical access to computing assets) or logical security (capability to login to a given system and application).

[SOURCE: IEC TS 62443-1-1:2009, 3.2.99]

3.3 Abbreviations

Abbreviations used in this document are shown in Table 2.

Table 2 – Abbreviations used in IEC 62061

CCF	Common Cause Failure(s)
DC	Diagnostic Coverage
EMC	Electromagnetic Compatibility
FVL	Full Variability Language
I/O	Input/Output
LVL	Limited Variability Language
HFT	Hardware Fault Tolerance
HW	Hardware
PFH, PFH_D	average frequency of dangerous failure per Hour
MRT	Mean Repair Time
MTTF	Mean Time To Failure
$MTTF_D$	Mean Time to Dangerous Failure
MTTR	Mean Time To Restoration
PFD	probability of dangerous failure on demand
PFD_{avg}	average probability of dangerous failure on demand
PL	Performance Level
PLC	Programmable Logic Controller
RDF	Ratio of Dangerous Failure
SFF	Safe Failure Fraction
SIL	Safety Integrity Level
SCS	Safety-Related Control System
SRS	Safety Requirements Specification
SW	Software

4 Design process of an SCS and management of functional safety

4.1 Objective

The objective of Clause 4 is to describe the design process and the tasks that have to be completed to realize each safety function performed by the related part of the control system for a given machine.

4.2 Design process

If as a result of the risk assessment of the whole machine according to ISO 12100 (see Figure 2), a need for risk reduction has been identified and if certain selected risk reduction measures depend on the control system, corresponding safety functions have to be specified.

NOTE 1 Examples of safety functions are given in ~~Annex H~~ Annex G.

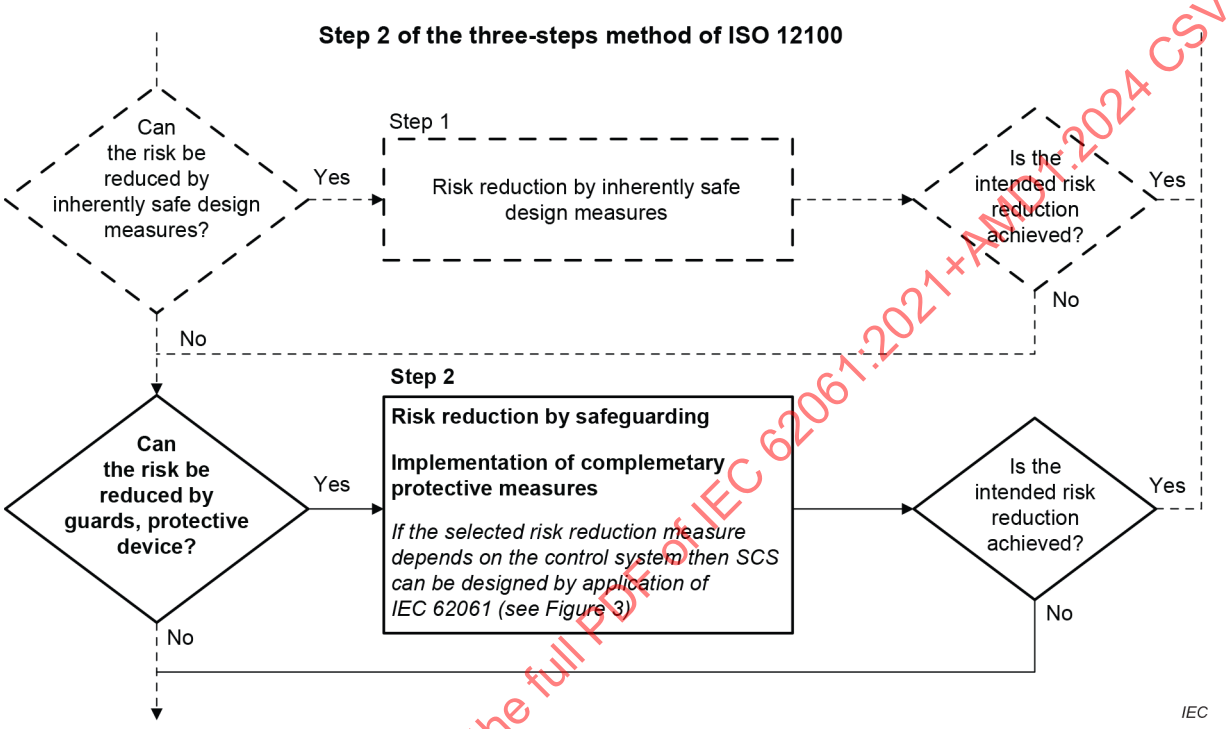
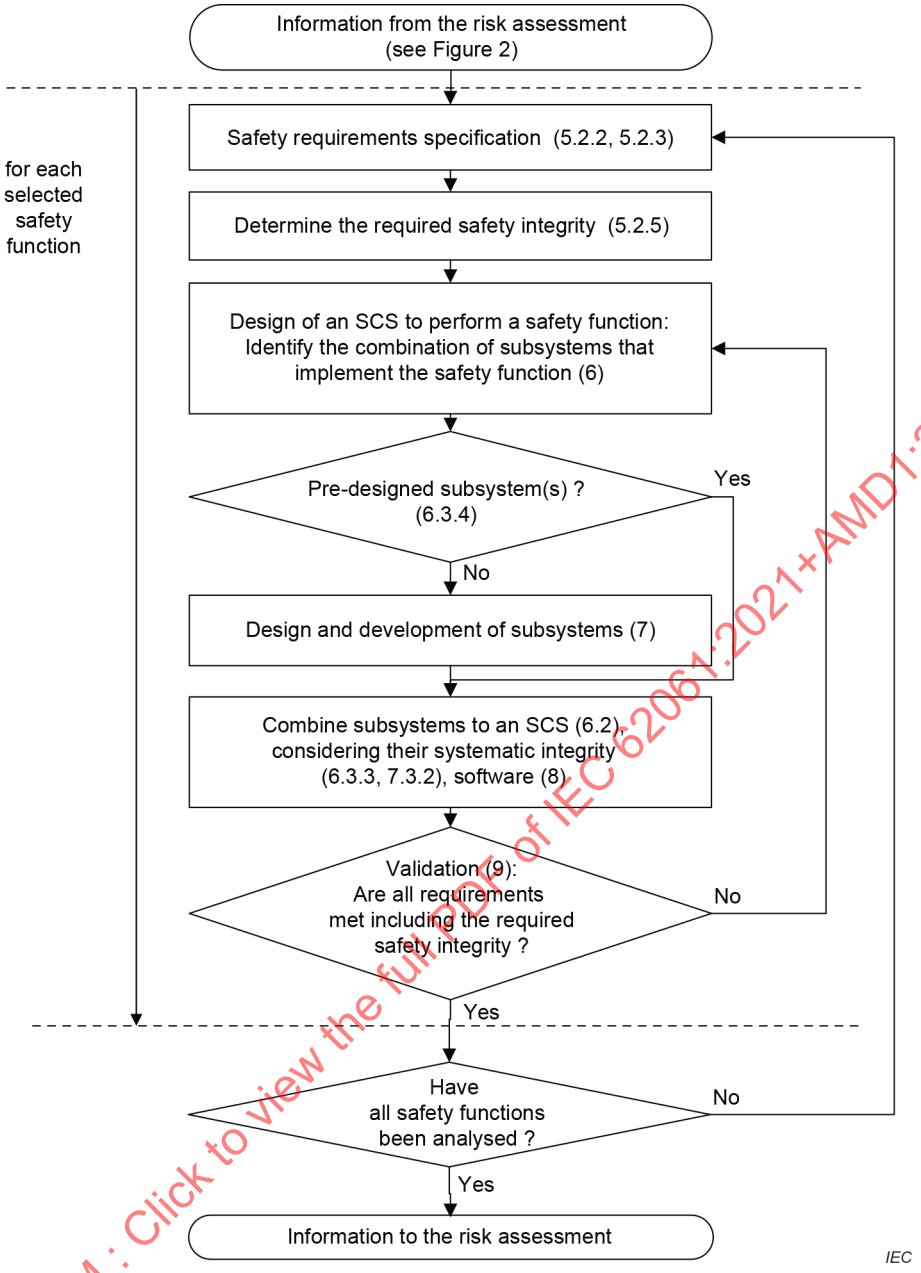


Figure 2 – Integration within the risk reduction process of ISO 12100 (extract)

NOTE 2 Figure 2 shows where the SCS contributes to the risk reduction process of ISO 12100: Step 2. The SCS supports the combined protective measures by the implementation of safety functions. ISO 12100 also provides general design rules for the machine which are applicable for the design of the SCS (see 6.2.11 and 6.2.12 of ISO 12100:2010).

The design process (see Figure 3) of each safety function implemented by a safety-related control system (SCS) shall include at least the safety function specification (see Clause 5) and the safety-related control system design (see Clause 6) and the associated verification and validation activities.



NOTE Each step described in the process flow diagram includes also verification activities.

Figure 3 – Iterative process for design of the safety-related control system

The realization of a safety function following the determined required safety integrity shall either be done by

– using an already developed SCS that meets the required safety integrity, or
– designing a new SCS using pre-designed subsystems according to Clause 6 or designing new subsystems according to Clause 7, or a combination of both.

If additional design considerations for software are necessary, Clause 8 applies.

A safety function can be implemented by one or more subsystem(s) of a safety-related control system (SCS), and several safety functions can share one or more subsystem(s) (e.g. a logic unit, power control element(s)), see examples in Figure 4. A control system can be subdivided into a safety-related part and a non-safety-related part. It is possible that one subsystem, which is involved in the implementation of safety functions, is also involved in the implementation of control functions. The designer may use any of the technologies available, singly or in combination.

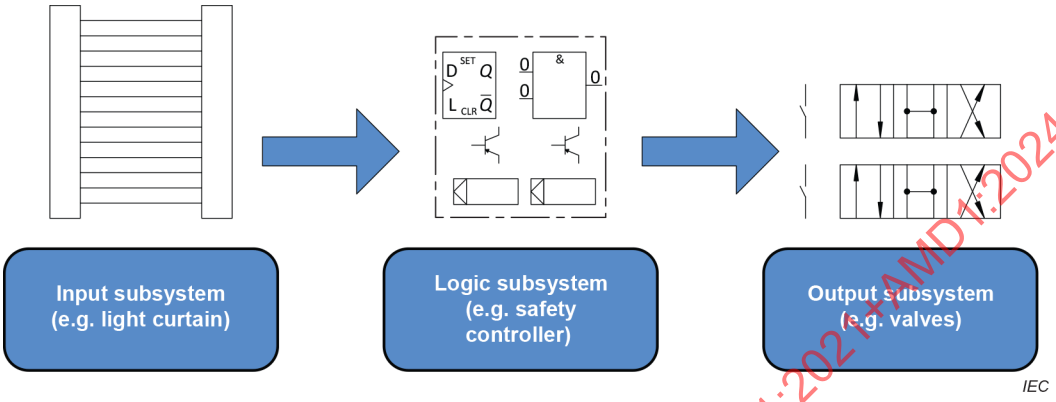


Figure 4 – Example of a combination of subsystems as one SCS

4.3 Management of functional safety using a functional safety plan

This subclause specifies management and technical activities that are necessary for the achievement of the required functional safety of the SCS.

NOTE 1 For further information, see IEC 61508-1:2010, Clause 6.

A functional safety plan shall be drawn up and documented for each SCS design project, and shall be updated as necessary. The functional safety plan is intended to provide measures for preventing incorrect specification, implementation, or modification issues.

The functional safety plan shall identify the relevant activities (see Figure 3) and shall be adapted to the project. See examples in Annex I.

NOTE 2 The functional safety plan can be part of a global machine design plan.

NOTE 3 The content of the functional safety plan depends upon the specific circumstances, which can include:

– size of project;
– degree of complexity;
– degree of novelty of design and technology;
– degree of standardization of design features;
– possible consequence(s) in the event of failure.

In particular, the functional safety plan shall:

a) identify the relevant activities specified in Clauses 5 to 9 and details of when they shall take place;
b) describe the policy and strategy to fulfil the specified functional safety requirements;
c) describe the strategy to achieve functional safety for the application software, results of a development, integration, verification and validation;
d) identify persons, departments or other units and resources that are responsible for carrying out and reviewing each of the activities specified in Clauses 5 to 9.

NOTE 4 The level of appropriate competency of the involved persons (i.e. training, technical knowledge, experience and qualifications) are taken into account. The appropriateness of competence is considered in relation to the particular application, taking into account all relevant factors including:

a) the responsibilities of the person;

b) the level of supervision required;

c) the potential consequences in the event of failure of the SCS;

d) the safety integrity levels of the SCS;

e) the novelty of the design, design procedures or application;

f) previous experience and its relevance to the specific duties to be performed and the technology being employed;

g) the type of competence appropriate to the circumstances (for example qualifications, experience, relevant training and subsequent practice, and leadership and decision-making abilities);

h) engineering knowledge appropriate to the application area and to the technology;

i) safety engineering knowledge appropriate to the technology;

j) knowledge of the legal and safety regulatory framework;

k) relevance of qualifications to specific activities to be performed.

e) identify or establish the procedures and resources to record and maintain information relevant to the functional safety of an SCS;

NOTE 5 The following are considered:

– the results of the hazard identification and risk assessment;

– the equipment used for safety-related functions together with its safety requirements;

– the organization responsible for maintaining functional safety;

– the procedures necessary to achieve and maintain functional safety (including SCS modifications).

f) describe the strategy for configuration management (see 4.4) taking into account relevant organizational issues, such as authorized persons and internal structures of the organization;

g) describe the strategy for modification (see 4.5);

h) establish a verification plan that shall include:

– details of when the verification shall take place;

– details of the persons, departments or units who shall carry out the verification;

– the selection of verification strategies and techniques;

– the selection and utilization of test equipment;

– the selection of verification activities;

– acceptance criteria; and

– the means to be used for the evaluation of verification results;

i) establish a validation plan comprising:

– results of previous verification;

– details of when the validation shall take place;

– identification of the relevant modes of operation of the machine (e.g. normal operation, setting);

– requirements against which the SCS shall be validated;

– the technical strategy for validation, for example analytical methods or statistical tests;

– acceptance criteria; and

– actions to be taken in the event of failure to meet the acceptance criteria.

NOTE 6 The validation plan indicates whether the SCS and its subsystems are to be subject to routine testing, type testing and/or sample testing.

4.4 Configuration management

The main operational aspects of configuration management are

– **identification** of the structure of the SCS, identifies e.g. system, subsystems, functions, function blocks, management documents, tools for creating a baseline;

– **controlling** of the release of an element created during each lifecycle phase at a specific point in time;

– **recording** and **reporting** of the status of each element which is and/or will be part of a baseline;

– **audit** and **review** of all elements and maintaining consistency among all elements of a baseline.

Procedures shall be developed for configuration management of the SCS during the overall, SCS system and software safety lifecycle phases, including in particular:

a) the point, in respect of specific phases, at which formal configuration control is to be implemented;

b) the procedures to be used for uniquely identifying all constituent parts of hardware and software;

c) the procedures for preventing unauthorized items from entering service.

The configuration management procedures shall be implemented in accordance with the functional safety plan (see 4.3).

The procedures for an appropriate change-control-process shall consider the requirements of procedures for defining a unique baseline of each version of the SCS.

4.5 Modification

If a modification is to be implemented, then relevant activities shall be identified specifically and an action plan shall be prepared and documented before carrying out any modification.

NOTE 1 The request for a modification can arise from, for example:

– safety requirements specification changed;

– conditions of actual use;

– incident/accident experience;

– change of material processed;

– obsolescence;

– modifications of the machine or of its operating modes.

NOTE 2 Interventions (e.g. adjustment, setting, repairs) on the SCS made in accordance with the information for use or instruction manual for the SCS are not considered to be a modification in the context of this subclause.

The reason(s) for the request for a modification shall be documented.

The effect of the requested modification shall be analysed to establish the effect on the safety function.

The modification impact analysis and the effect on the functional safety of the SCS shall be documented.

All accepted modifications that have an effect on the SCS shall initiate a return to an appropriate design phase for its hardware and/or for its software (e.g. specification, design, integration, installation, commissioning, and validation). All subsequent phases and management procedures shall then be carried out in accordance with the procedures specified for the specific phases in this document. All relevant documents shall be revised, amended and reissued accordingly.

5 Specification of a safety function

5.1 Objective

This clause sets out the procedures to specify the requirements of safety function(s) to be implemented by the SCS.

5.2 Safety requirements specification (SRS)

5.2.1 General

Each safety function shall be specified by:

– functional requirements specification (see 5.2.3);
– safety integrity requirements specification (see 5.2.5)

and these shall be documented in the safety requirements specification (SRS).

Where a product standard specifies the safety requirements for the design of an SCS or subsystem (e.g. ISO 13851 for two-hand control devices), these should be considered.

5.2.2 Information to be available

The following information shall be used to produce both the functional requirements specification and safety integrity requirements specification of SCS:

– results of the risk assessment for the machine including all safety functions determined to be necessary for the risk reduction process for each specific hazard;
– machine operating characteristics, including:
  - modes of operation of machine,
  - cycle time,
  - response time performance,
  - environmental conditions,
  - interaction of person(s) with the machine (e.g. repairing, setting, cleaning);
– all information relevant to the safety function(s) which can have an influence on the SCS design including, for example:
  - a description of the behaviour of the machine that a safety function is intended to achieve or to prevent;
  - all interfaces between the safety functions, and between safety functions and any other function (either within or outside the machine);
  - required fault reaction functions of the safety function.

NOTE Some of the information might not be available or sufficiently defined before starting the iterative design process of SCS, so the SCS safety requirements specifications can be required to be updated during the design process.

5.2.3 Functional requirements specification

The functional requirements specification shall describe details of each safety function to be performed including as applicable:

– a description of each safety function;
– the condition(s) (e.g. operating mode) of the machine in which the safety function shall be active, disabled, configured or parameterized;
– the priority of those functions that can be simultaneously active and that can cause conflicting action;
– the reset of a safety function;
– the frequency of operation of each safety function (rate of operating cycles, duty cycle);
– demand mode of operation;

NOTE 1 For definitions refer to 3.2.26, 3.2.27, 3.2.28.

– the required response time of each safety function;
– the interface(s) of the safety functions to other machine functions;

NOTE 2 This could include a description of methods intended to give status information to users of the machinery.

– a description of fault reaction function(s) and any constraints on, for example, re-starting or continued operation of the machine in cases where the initial fault reaction is to stop the machine;
– tests and any associated facilities (e.g. test equipment, test access ports);
– a description of the operating environment (e.g. electromagnetic immunity, temperature, humidity, dust, chemical substances, mechanical vibration and shock);

NOTE 3 The specification of the electromagnetic environmental condition is within the scope of IEC 61000-1-2. The electromagnetic environment is defined as the totality of electromagnetic phenomena existing at a particular location. These phenomena can vary over time.

The electromagnetic environment is influenced by, for example:

– fixed and moving sources of electromagnetic energy,
– low, medium and high voltage equipment,
– control, signalling, communication and power systems,
– intentional radiators,
– physical processes (e.g. atmospheric discharges, switching actions),
– random or infrequent transients,

which all can produce disturbances that adversely impact the safety-related system or element under consideration.

– rate of operating cycles, duty cycle, and/or utilisation category, for devices intended for use in the safety function;

NOTE 4 The duty cycle of subsystems or subsystem elements can be higher than required for the safety function, e.g. when used also for non-safety-related machine functions (the total number of cycles is to be considered).

– other specific requirements which can impact functional safety.

5.2.4 Estimation of demand mode of operation

The demand mode of operation shall be estimated by applying the respective definitions. While low demand mode operation is possible for a safety function, this document concentrates on high demand and continuous mode. When demand rate is estimated to be low, a high demand mode can be assumed by activation of the safety function at least once per year. Then apply this document for the design. This is a straightforward application of the definition and shown in Figure 5 as a workflow.

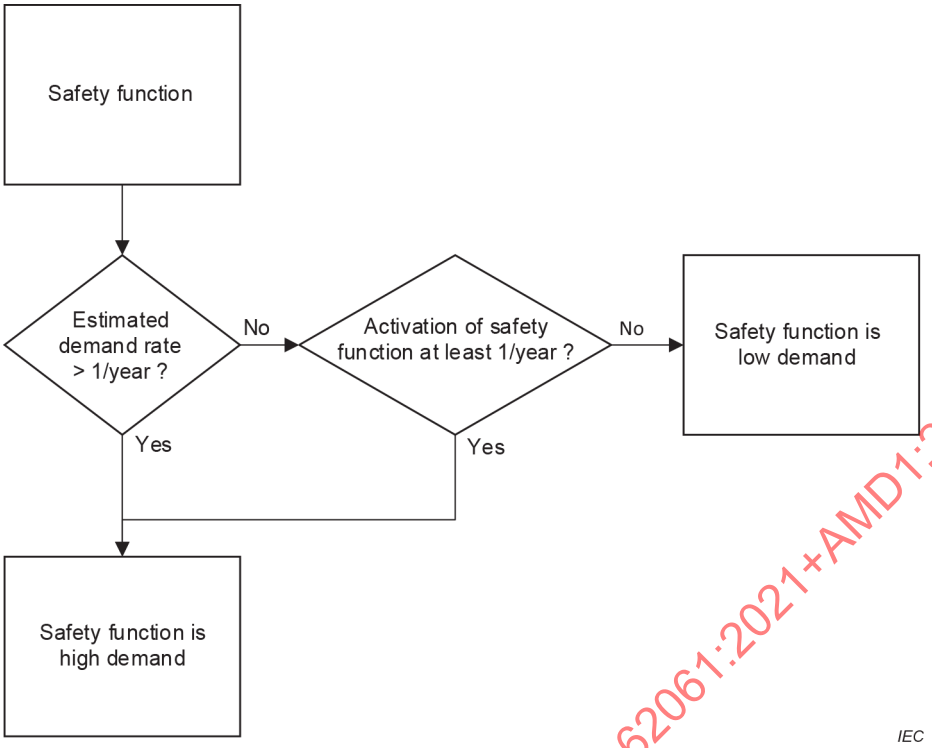


Figure 5 – By activating a low demand safety function at least once per year it can be assumed to be high demand

5.2.5 Safety integrity requirements specification

The safety integrity requirements for each safety function shall be derived from the risk assessment to ensure the necessary risk reduction can be achieved. In this document, a safety integrity requirement is expressed as a target failure measure for the PFH.

The required safety integrity for each safety function to be carried out by an SCS shall be specified in terms of SIL according to Table 3 and documented.

Table 3 – SIL and limits of PFH values

SIL	Limits of PFH values (1/h)
1	$< 10^{-5}$
2	$< 10^{-6}$
3	$< 10^{-7}$

The determination of the required safety integrity is the result of the risk assessment and refers to the amount of the risk reduction to be carried out by the SCS. Examples of a methodology are given in Annex A.

NOTE 1 Where a product standard specifies a required SIL for a safety function then this takes precedence over Annex A.

NOTE 2 Further guidance on relationship between risk assessment according to ISO 12100 and product standards is provided in ISO TR 22100-1.

6 Design of an SCS

6.1 General

The SCS shall be designed in accordance with the safety requirements specification (see 5.2), using one or several subsystems by:

– selection of subsystems (see 6.2, 6.3 and Clause 7);
– determining the safety integrity (see 6.4);
– complying to the requirements of the systematic safety integrity of the SCS (see 6.5), including, where applicable, electromagnetic immunity (see 6.6), security (see 6.8), periodic testing (see 6.9) and, software (see 6.7 and Clause 8).

6.2 Subsystem architecture based on top down decomposition

The following Clause 6 describes the design process of an SCS. An SCS can include:

– one or several pre-designed subsystem(s), and/or
– one or several subsystem(s) developed according to this document, based on subsystem element(s) (see Clause 7).

NOTE 1 The designer of a pre-designed subsystem can be a manufacturer of the machine or a device manufacturer.

NOTE 2 The relevant safety integrity characteristic values come from the designer of pre-designed subsystem.

6.3 Basic methodology – Use of subsystem

6.3.1 General

Each safety function identified in the risk reduction process (see Clause 4) is performed by an SCS consisting of one or several subsystems. A failure of any subsystem will result in the loss of the whole safety function. Subclause 6.2 describes the principle of this allocation task.

Where an SCS or part of an SCS (i.e. its subsystem(s)) is to implement both safety functions and other functions, then all its hardware and software shall be treated as safety-related unless it can be shown that the implementation of the safety functions and other functions is sufficiently independent (i.e. that the normal operation or failure of any other functions do not affect the safety functions).

NOTE 1 Sufficient independence of implementation is established by showing that the probability of a dependent failure between the non-safety and safety-related parts is equivalent to that of the safety integrity level of the SCS. IEC 61508-3:2010, Annex F describes techniques for achieving non-interference between software elements.

For an SCS or its subsystems that implements safety functions of different safety integrity levels, its hardware and software shall be treated as requiring the highest safety integrity level unless it can be shown that the implementation of the safety functions of the different safety integrity levels is sufficiently independent.

NOTE 2 Sufficient independence of implementation is established by showing that the probability of a dependent failure between the non-safety and safety-related parts is sufficiently low in comparison with the highest safety integrity level associated with the safety functions involved.

Where digital data communication is used as a part of an SCS implementation, it shall satisfy the relevant requirements of IEC 61508-2:2010, 7.4.11 (which refers to IEC 61784-3 (all parts) for functional safety fieldbuses) in accordance with the SIL target(s) of the safety function(s).

6.3.2 SCS decomposition

Each safety function shall be decomposed to a structure of sub-function(s). The decomposition process shall lead to a structure of sub-functions that fully describes the functional and integrity requirements of the SCS. This process should be applied down to that level that permits the functional and integrity requirements determined for each sub-function to be allocated to a single subsystem.

Figure 6 shows examples of typical decompositions starting with a detection and evaluation of an ‘initiation event’ and is ending with an output causing a reaction of a ‘machine actuator’.

For each sub-function the following shall be specified:

– the safety requirements (functional and integrity), and
– inputs and outputs of each sub-function.

NOTE 1 The inputs and outputs of each sub-function are the information that is transferred, for example speed, position, mode of operation, etc.

NOTE 2 The sub-functions can have associated diagnostic functions (see 7.4.3.3, diagnostic coverage).

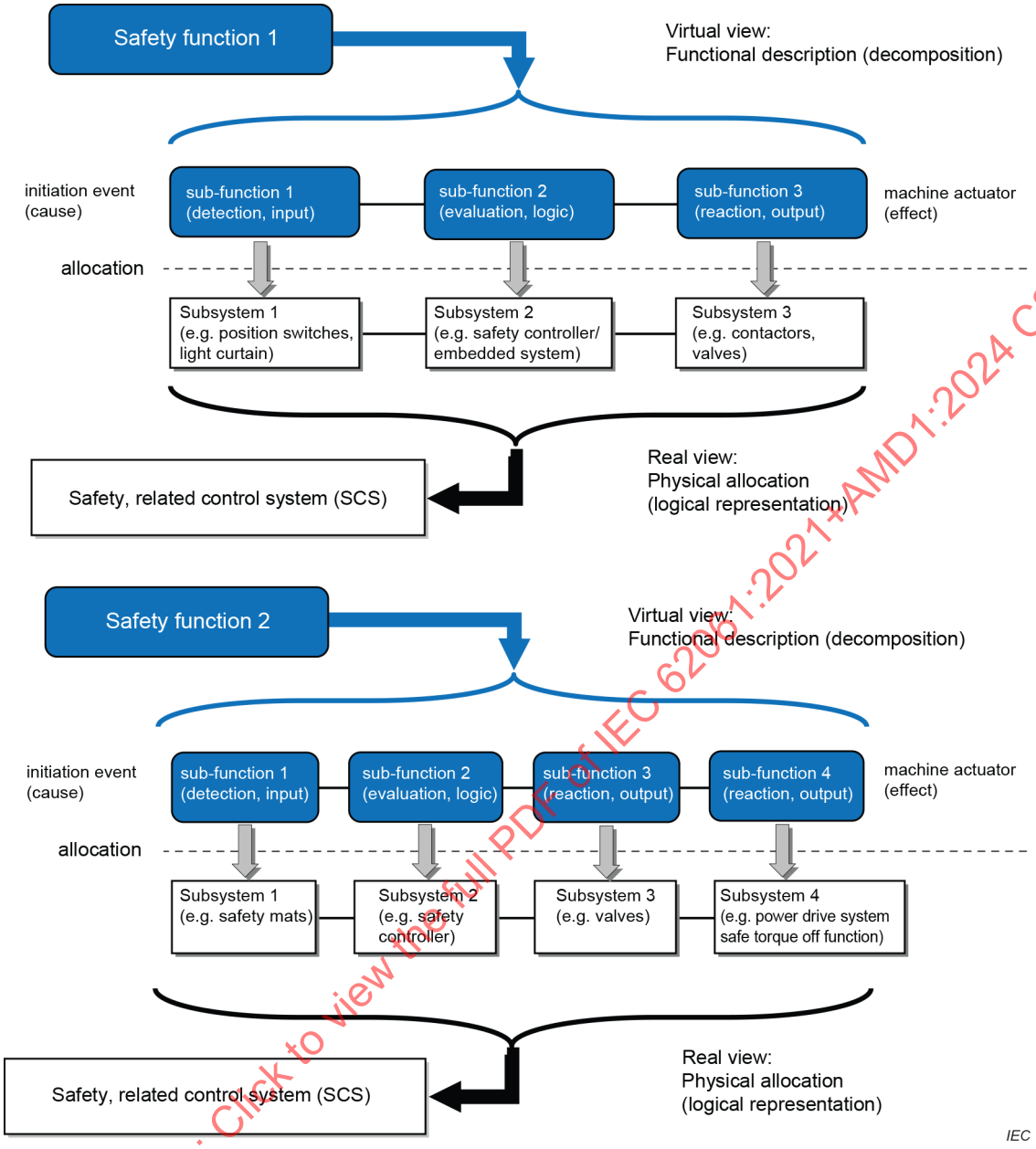
NOTE 3 An SCS can consist of one single subsystem. Example for an SCS implementation with a single subsystem is an “Intelligent” sensor unit (e.g. laser scanner) with integrated output switching device (e.g. relay).

NOTE 4 A subsystem which implements a sub-function can consist of more than one physical unit. An example is a safety controller which has separate input, logic, output (and safety-related fieldbus communication) units. The manufacturer can provide separately the safety-related data for the units.
Another example is a safety relay module which monitors the status of an input device. When the safety relay module does not contain enough output contacts for the specific sub-function then an extension safety module can be added. The manufacturer(s) provides separately the safety-related data for all the modules.

NOTE 5 When decomposing safety requirements into sub-requirements, proper documentation and configuration management processes are conducted for ensuring the maintenance of bi-directional traceability between decomposed requirements.

The decomposition of an SCS into subsystems represented in Figure 6 is typical but the whole SCS can be realized by any number of subsystems.

Figure 6 does not present the possible diagnostic functions that can be required to fulfil the safety requirements.



NOTE 1 The fieldbus communication can be part of one or more subsystems.

NOTE 2 Interconnection (e.g. wiring) aspects can be relevant in subsystem(s) (see 7.3.2.2).

Figure 6 – Examples of typical decomposition of a safety function into sub-functions and its allocation to subsystems

6.3.3 Sub-function allocation

Each sub-function shall be allocated to a subsystem within the architecture of the SCS. More than one sub-function (for example implementing different safety functions), can be allocated to a subsystem.

NOTE An example of a subsystem that implements several sub-functions is a safety controller which acts as a logic solver for guard interlocking function and overspeed protection function.

6.3.4 Use of a pre-designed subsystem

The safety performance of a pre-designed subsystem, according to other standards, shall be in line with Table 4.

Table 4 – Required SIL and *PFH* of pre-designed subsystem

IEC 62061 (IEC 61508)	IEC 62061	IEC 61508 [a]	ISO 13849 [b]	IEC 61496
PFH	**SIL**	**at least ...**	**at least ...**	**at least ...**
$< 10^{-5}$	SIL 1	SIL 1	PL b, c	Type 2
$< 10^{-6}$	SIL 2	SIL 2	PL d	Type 3
$< 10^{-7}$	SIL 3	SIL 3	PL e	Type 4
NOTE A relation between IEC 62061 and IEC 61511 (all parts) or ISO 26262 cannot be assumed within this table.				
[a] This column includes SIL-based standards that fulfil the architectural constraints of IEC 61508, such as IEC 61800-5-2 and IEC 60947-5-3. [b] Does not apply to subsystems using complex components, unless they meet the requirements of IEC 61508 or applicable functional safety products standards. Performance Level b does not correspond to SIL1 in case of a category B (ISO 13849-1) structure.				

6.4 Determination of safety integrity of the SCS

6.4.1 General

The SIL(s) that can be achieved by the SCS shall be considered separately for each safety function and shall be determined from the SIL and the *PFH* of each subsystem, as follows:

– the SIL that is achieved is equal to or less than the lowest SIL of any of the subsystems, and

– the SIL is limited by the sum *PFH* value of all subsystems according to Table 3.

Figure 7 shows an example of an SCS with safety integrity of SIL 2 despite the overall *PFH* value being suitable for a higher SIL.

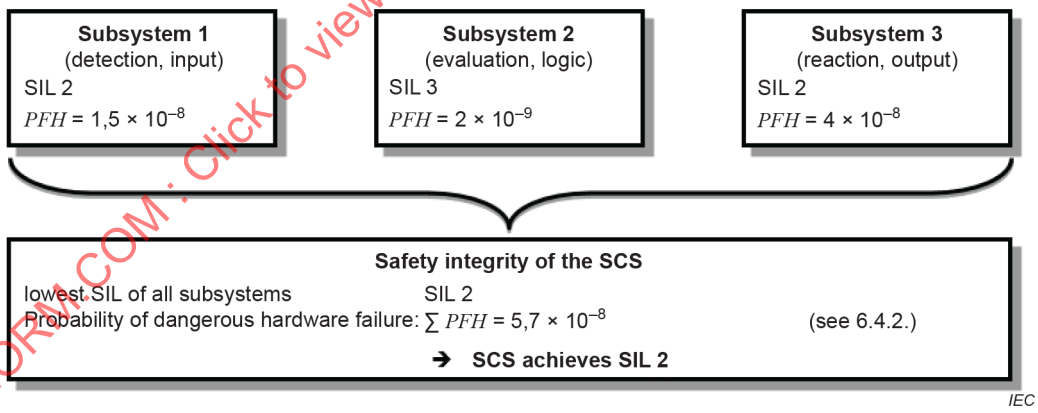


Figure 7 – Example of safety integrity of a safety function based on allocated subsystems as one SCS

NOTE An SCS can be a combination of subsystems based on different architectures.

6.4.2 PFH

The *PFH* of each safety function due to dangerous random hardware failures shall be equal to or lower than the *PFH* of Table 3 related to required SIL as specified in the safety requirements specification.

The estimation of the PFH shall be based on the PFH of each relevant subsystem including, where appropriate, for digital data communication processes between subsystems. The PFH of the SCS is the sum of the probabilities of dangerous random hardware failure of all subsystems involved in the performance of the safety function and shall include, where appropriate, the maximum probability of dangerous transmission errors (P_{TE}) for digital data communication:

$$PFH = PFH_1 + \cdots + PFH_n + P_{TE} \quad (2)$$

NOTE 1 This approach is based on the definition of a subsystem which states that a failure of any subsystem will result in a failure of the SCS (see 6.3.1).

NOTE 2 Hardware wiring aspects are part of systematic integrity and possible failures can be detected by diagnostics.

NOTE 3 For the determination of the P_{TE}, see for example IEC 61784-3.

6.5 Requirements for systematic safety integrity of the SCS

6.5.1 Requirements for the avoidance of systematic hardware failures

The following measures shall apply when appropriate:

a) the SCS shall be designed and implemented in accordance with the functional safety plan (see 4.3);

b) proper selection, combination, arrangements, assembly and installation of subsystems, including cabling, wiring and any interconnections. Wiring interconnection of subsystems may require specific fault considerations and fault exclusions (see 7.3.3);

c) use of the SCS within the manufacturer's specification;

d) use of subsystems that have compatible operating characteristics;

NOTE See also ISO 13849-2:2012, Annexes A, B, C and D.

e) the SCS shall be installed and protected in accordance with IEC 60204-1, including earth fault detection;

f) undocumented modes of component operation shall not be used (e.g. 'reserved' registers of programmable equipment);

g) consideration of foreseeable misuse, environmental changes or modification(s);

h) manufacturer's instructions (including e.g. application examples) of both interconnected subsystems (outputs of the preceding subsystem and inputs of the subsequent subsystem) shall be applied; these can include:

- hardware aspects (e.g. interface information, shielding, signal level, pressure threshold, test pulses, architectural constraints),
- software aspects (e.g. definition of data communication telegrams), and
- diagnostic coverage aspects.

In addition, at least one of the following techniques and/or measures shall be applied taking into account the complexity of the SCS and the SIL(s) for those functions to be implemented by the SCS:

i) SCS hardware design review (e.g. by inspection or walk-through): to reveal by reviews and/or analysis any discrepancies between the specification and implementation;

NOTE 1 In order to reveal discrepancies between the specification and implementation, any points of doubt or potential weak points concerning the realization, the implementation and the use of the product are documented so they can be resolved, taking into account that on an inspection procedure the author is passive and the inspector is active whilst on a walk-through procedure the author is active and the inspector is passive.

j) advisory tools such as computer-aided design packages capable of simulation or analysis, and/or the use of computer-aided design tools to perform the design procedures systematically with the use of pre-designed elements that are already available and tested;

NOTE 2 The integrity of these tools can be demonstrated by specific testing, or by an extensive history of satisfactory use, or by independent verification of their output for the particular SCS that is being designed.

k) simulation: perform a systematic and complete assimilation of an SCS design in terms of both functional performance and the correct dimensioning and interaction of its subsystems.

EXAMPLE The functions of the SCS can be simulated on a computer via a software behavioural model where individual subsystems or subsystem elements each have their own simulated behaviour, and the response of the circuit in which they are connected is examined by looking at the marginal data of each subsystem or subsystem element.

6.5.2 Requirements for the control of systematic faults

The following measures shall be applied:

a) use of de-energization: the SCS shall be designed so that with loss of its supply, a safe state of the machine is achieved or maintained;

b) measures to control the effect of temporary subsystem failures: the SCS shall be designed so that, for example:

- supply variation (e.g. interruptions, dips) to an individual subsystem or a part of a subsystem does not lead to a hazard (e.g. a voltage interruption that affects a motor circuit shall not cause an unexpected start-up when the supply is restored), and

NOTE 1 See also relevant requirements of IEC 60204-1. In particular:

– overvoltage or undervoltage can be detected early enough so that all outputs can be switched to a safe condition by the power-down routine or a switch-over to a second power unit; and/or

– where necessary, overvoltage or undervoltage can be detected early enough so that the internal state can be saved in non-volatile memory, so that all outputs can be set to a safe condition by the power-down routine, or all outputs can be switched to a safe condition by the power-down routine or a switch-over to a second power unit.

See also relevant information in IEC 61131-2.

- the effects of electromagnetic interference from the physical environment or a subsystem(s) do not lead to a hazard;

c) measures to control the effects of errors and other effects arising from any data communication, including transmission errors (such as repetitions, deletion, insertion, re-sequencing, corruption, delay and masquerade);

NOTE 2 Further information can be found in IEC 61784-3:~~2016~~2021, Table 1 and IEC 61508-2:2010, 7.4.11.2.

NOTE 3 The term 'masquerade' means that the true contents of a message are not correctly identified. For example, a message from a non-safety component is incorrectly identified as a message from a safety component.

d) when a dangerous fault occurs at an interface, the fault reaction function shall be performed before the hazard due to this fault can occur. When a fault that reduces the hardware fault tolerance to zero occurs, this fault reaction shall take place before the estimated MTTR (see 3.2.39) is exceeded.

The requirements of item d) apply to interfaces that are inputs and outputs of subsystems and all other parts of subsystems that include or require cabling during integration (for example output signal switching devices of a light curtain, output of a guard position sensor).

NOTE 4 This does not require that a subsystem or subsystem element on its own has to detect a fault on its outputs(s). The fault reaction function can also be initiated by any subsequent subsystem after a diagnostic test is performed.

6.6 Electromagnetic immunity

The function of electrical or electronic safety-related systems shall not be affected by external influences in a way that could lead to an unacceptable risk. Acceptable performance with respect to electromagnetic disturbances is therefore mandatory. A comprehensive safety analysis shall include the effects of electromagnetic disturbances and the electromagnetic immunity limits that are required to achieve functional safety. These limits should be derived taking into account both the electromagnetic environment and the required safety integrity levels.

The SCS shall fulfil the applicable requirements of IEC 61000-1-2.

NOTE 1 The appropriate immunity levels in the case of industrial environments are given by IEC 61326-3-1 or IEC 61000-6-7 as a minimum.

NOTE 2 If a subsystem has been designed following an appropriate safety-related product standard (e.g. IEC 61496-1, etc.) or to IEC 61326-3-1 or IEC 61000-6-7, it can be possible that information is supplied with the subsystem that facilitates verification of the SCS level requirements by analysis.

NOTE 3 Guidance design principles are available in EMC standards, but functional safety standards require higher immunity levels. It is important to recognise that higher immunity levels, or additional immunity requirements, than those specified in such standards can be necessary for particular locations or when the equipment is intended for use in harsher, or different, electromagnetic environments.

6.7 Software based manual parameterization

6.7.1 General

Some safety related subsystems or SCS need parameterization to carry out a safety function or a sub-function. For example, a converter with integrated sub-functions has to be parameterized via a PC-based configuration tool, with respect to the upper safe speed limit. Similarly, to properly establish the detection zone of a laser scanner, parameters such as angle and distance can need to be configured per the manufacturer's safety documentation and the machine risk assessment.

The objective of the requirements for software based manual parameterization is to guarantee that the safety-related parameters specified for a safety function or a sub-function are correctly transferred into the hardware performing the safety function or a sub-function. Different methods can be applied to set such parameters; even dip switch based parameterization can be used to set or change safety-related parameters. However, PC-based tools with dedicated parameterization software, commonly called configuration or parameterization tools, are becoming more prevalent. This subclause is limited in scope to only manual, software based parameterization that is performed and controlled by an authorized person.

NOTE 1 Safety-related parameterization which is carried out automatically without human interaction, for example, based on input signals, is not considered in this Subclause 6.7.

NOTE 2 Direct control of a machine by an operator, e.g. speed control of a forklift truck is not considered as manual parameterization as described in this subclause.

NOTE 3 If the configuration or parameterization tool is pre-designed in accordance with IEC 61508-3, for example together with its dedicated subsystem, it is assumed that there will be no dangerous failures due to the influences listed in 6.7.2 or any other influence that is reasonably foreseeable. The requirements of 6.7.5 apply when a software based manual parameterization is performed with the pre-designed tool.

6.7.2 Influences on safety-related parameters

During software based manual parameterization, the parameters can be affected by several influences, such as:

– data entry errors by the person responsible for parameterization;
– faults of the software of the parameterization tool;
– faults of further software and/or service provided with the parameterization tool;

– faults of the hardware of the parameterization tool;
– faults during transmission of parameters from the parametrization tool to the SCS or a subsystem;
– faults of the SCS or a subsystem to store transmitted parameters correctly;
– systematic interference during the parameterization process, e.g. by electromagnetic interference or loss of power;
– interference due to external influences or factors, such as electromagnetic interference or (random) loss of power.

With no measures applied to counteract, avoid or control potential dangerous failures caused by the influences listed above, such influence can lead to the following:

– parameters are not updated by the parameterization process, completely or in parts without notice to the person responsible for the parametrization;
– parameters are incorrect, completely or in parts;
– parameters are applied to an incorrect device, such as when transmission of parameters is carried out via a wired or wireless network.

6.7.3 Requirements for software based manual parameterization

Software based manual parameterization shall use a dedicated tool provided by the manufacturer or supplier of the SCS or the related subsystem(s). This tool shall have its own identification (name, version, etc.). The SCS or the related subsystem(s) and the parameterization tool shall have the capability to prevent unauthorized modification, for example by using a dedicated password.

Parameterization while the machine is running shall be permitted only if it does not cause a hazardous situation.

When using a pre-designed SCS or subsystem that is capable of software based manual parameterization, the objective is to prevent dangerous failure due to the influences listed in 6.7.2 or any other influence that is reasonably foreseeable.

It is possible to fulfil the requirements by using a pre-designed SCS or subsystem, or the design of the SCS or subsystem shall follow this document. Aspects of parametrization shall be included in the validation of the SCS.

The following requirements shall be fulfilled.

a) The design of the software based manual parameterization shall be considered as a safety-related aspect of SCS design that is described in a safety requirements specification, e.g. the software safety requirements specification (see 8.3.2.2 and 8.4.2.2).
b) The SCS or subsystem shall provide means to check the data plausibility, e.g. checks of data limits, format and/or logic input values.
c) The integrity of all data used for parameterization shall be maintained. This shall be achieved by applying measures to
   - control the range of valid inputs;
   - control data corruption before transmission;
   - control the effects of errors from the parameter transmission process;
   - control the effects of incomplete parameter transmission;
   - control the effects of faults and failures of hardware and software of the parameterization; and
   - control the effect of interruption of the power supply.

d) The parameterization tool shall fulfil all relevant requirements for a subsystem according to IEC 61508 to ensure correct parameterization.

e) Alternatively to d) a special procedure shall be used for setting the safety-related parameters. This procedure shall include confirmation of input parameters to the SCS by either:

- retransmitting of modified parameters to the parameterization tool; or
- other means to confirm the integrity of the parameters

as well as subsequent confirmation, for example by a suitably skilled person and by means of an automatic check by a parameterization tool. New values of safety-related parameters shall not be activated before the changes are acknowledged and confirmed.

NOTE This is of particular importance where a parameterization software tool uses a device not specifically intended for this purpose (e.g. personal computer or equivalent).

The software modules used for encoding/decoding within the transmission/retransmission process and software modules used for visualization of the safety-related parameters to the user shall, as a minimum, use diversity in function(s) to avoid systematic failures.

6.7.4 Verification of the parameterization tool

As a minimum, the following verification activities shall be performed to verify the basic functionality of the parameterization tool:

– verification of the correct setting for each safety-related parameter (minimum, maximum and representative values);

– verification that the safety-related parameters are checked for plausibility, for example by detection of invalid values, etc.;

– verification that means are provided to prevent unauthorized modification of safety-related parameters.

NOTE This is of particular importance where the parameterization is carried out using a device not specifically intended for this purpose (e.g. personal computer or equivalent).

6.7.5 Performance of software based manual parameterization

Software based manual parameterization shall be carried out using the dedicated parameterization tool provided by the manufacturer or supplier of the SCS or the related subsystem(s) and shall be documented according to the requirements given in the information for use. This information can originate from different parties, see also 10.3 (information for use). Protective measures against unauthorized access shall be activated and used.

The initial parameterization, and subsequent modifications to the parameterization, shall be documented. The documentation shall include:

a) the date of initial parameterization or change;

b) data or version number of the data set;

c) name of the person carrying out the parameterization;

d) an indication of the origin of the data used (e.g. pre-defined parameter sets);

e) clear identification of safety related parameters;

f) clear identification of the SCS which are subject to specific parametrization settings.

6.8 Security aspects

Security covers intentional attacks on the hardware, application programs and related software, as well as unintended events resulting from human error.

NOTE 1 Security aspects are considered in the security lifecycle of the machine (or higher system level) and throughout the life cycle of the machine.

NOTE 2 Since this document does not provide specific requirements on security aspects, guidance is provided in IEC ~~TR~~ TS 63074, ISA TR84.00.09, ISO/IEC 27001:~~2013~~2022, ISO TR 22100-4 and IEC 62443 (all parts).

When security countermeasures are applied, they shall not adversely affect safety integrity (e.g. increase in response time, etc.). This can require an iterative multi-disciplinary team analysis.

When security countermeasures implemented within the SCS are declared, then information shall be provided as appropriate.

6.9 Aspects of periodic testing

Periodic testing of the safety function or sub-functions serves two different purposes:

– periodic testing confirms at a given point of time that the tested function is not failed;
– periodic testing in conjunction with inspections assures that the boundary conditions for equipment reliability figures are met.

In general, two types of periodic testing are distinguished:

– diagnostic tests are carried out automatically (initiated automatically or manually) and frequently (related to the process safety time and demand rate);

NOTE 1 Periodic testing can apply to a sub-function or a safety function.

– periodic tests try to verify the complete function, typically by simulating the dangerous condition to the sensors or at least to the logic solver. Also, inspections for ageing and degradation of components are done as part of proof tests.

NOTE 2 The dangerous failures that cannot be detected by the diagnostics are considered to be undetected dangerous failures (related failure rate λ_{DU}). These failures can only be found by the proof-test.

In order to use periodic testing as safety integrity assurance, the following conditions shall be met:

– in the test procedure, a fault reaction shall be implemented to set the relevant parts of the machine in a safe state as consequence of a detected fault;

NOTE 3 The nature of fault reaction can be different for diagnostic and proof test and this also depends on the demand mode and architecture. For architecture of functions with HFT 0 and high or continuous demand, it is usually required to immediately shut-down the machinery.

– the test interval shall be adequate to reveal failures in respect to demand rate;
– for diagnostic tests, see also 7.4.3 for specific requirements.

7 Design and development of a subsystem

7.1 General

The subsystem shall be designed in accordance with its safety requirements specification (see 5.2), including basically:

– the functional requirements;
– the requirements for hardware safety integrity:
  • architectural constraints (see 7.4) and
  • *PFH* (see 7.6);
– the requirements for systematic integrity (see 7.3.2 and estimation of CCF in Annex E);
– the requirements for subsystem behaviour on detection of a fault (fault reaction) (see 7.4.3);
– the requirements for software (see Clause 8).

The following information of Table 5 shall be available where relevant for each subsystem during the design and development.

Table 5 – Relevant information for each subsystem

Functional description	
1)	A functional description of the function(s) and interface(s) of the subsystem
Hardware information	
2)	The estimated rates of failure (due to random hardware failures and failure modes) for each subsystem element which could cause a dangerous failure of the subsystem (see Annex C)
3)	Any test and/or maintenance requirements
4)	The probability of dangerous communication errors for digital data communication processes, where applicable
Environmental conditions	
5)	The environment and operating conditions which should be observed in order to maintain the validity of the estimated rates of failure due to random hardware failures
6)	The useful lifetime (see 7.3.4.2) of the subsystem which should not be exceeded, in order to maintain the validity of the estimated rates of failure due to random hardware failures
Design information	
7)	The diagnostic coverage and/or safe failure fraction and the diagnostic test interval (see 7.4.3 and 7.4.4)
8)	Limits on the application of the subsystem which should be observed in order to avoid or control systematic failures
9)	Information which is required to identify the hardware and software configuration of the subsystem
10)	The highest SIL that can be claimed for a safety function under consideration which uses the subsystem on the basis of: – architectural constraints, – measures and techniques used to avoid or control systematic faults being introduced during the design and implementation of the hardware and software of the subsystem, and – the design features that make the subsystem tolerant against systematic faults. NOTE One subsystem can implement sub-functions of several safety functions with different SIL.

7.2 Subsystem architecture design

The architecture of a subsystem is defined by a process of functional decomposition similar as that of the complete safety function that leads to the SCS architecture – see 6.3.2: The specific sub-function of the subsystem can be decomposed into sub-functions of the next lower order which are then assigned to subsystem elements.

As a result, a set of subsystem element(s) can be defined that meets the functional requirements and the integrity requirements of the sub-function.

NOTE 1 A subsystem can be designed by using one single subsystem element.

NOTE 2 The decomposition into subsystem element(s) can be an iterative process.

NOTE 3 The failure of a subsystem element does not necessarily result in a failure of the subsystem or sub-function. Where subsystem elements are parts of redundant channels, a single element failure will not result in a failure of the safety function.

The design of the subsystem architecture shall be documented in terms of its subsystem elements and their interrelationships, e.g. circuit diagram with description, safety-related block diagram.

Subsystem(s) incorporating complex components shall comply with appropriate product standards or IEC 61508-2 and IEC 61508-3 as appropriate for the required SIL and the design shall use Route 1_H (see IEC 61508-2:2010, 7.4.4.2) for high demand and\or continuous mode.

Where a subsystem design includes such a complex component as a subsystem element, it can be considered as a low complexity component in the context of a subsystem design since its relevant failure modes, behaviour on detection of a fault, rate of failure, and other safety-related information are known. Such components shall only be used in accordance with its specification and the relevant information for use provided by its manufacturer.

NOTE 4 In this document, it is presumed that the design of complex programmable electronic subsystems or subsystem elements conforms to the relevant requirements of IEC 61508 and uses Route 1_H (see IEC 61508-2:2010, 7.4.4.2).

7.3 Requirements for the selection and design of subsystem and subsystem elements

7.3.1 General

There are two types of requirements to subsystems and subsystem elements:

– qualitative requirements: systematic integrity; fault consideration(s) and fault exclusion(s);

– quantitative requirements: failure rate and other relevant parameters.

Qualitative requirements are defined in the following Subclauses 7.3.2 and 7.3.3. Where not explicitly stated otherwise, these requirements apply independently of the SIL requirement to the safety function from SIL 1 up to SIL 3.

NOTE SIL 4 is not considered in this document, as it is not suitable to the risk reduction requirements associated with machinery. For requirements applicable to SIL 4, see IEC 61508-1 and IEC 61508-2.

The quantitative requirements are described in 7.4 in general terms and for determination of the PFH, refer to 6.3.2 and 7.6.

7.3.2 Systematic integrity

7.3.2.1 General

The systematic safety integrity requirements for a subsystem are met by fulfilling the requirements in 7.3.2.2 and 7.3.2.3 and are the same for SIL 1, SIL 2 and SIL 3.

NOTE The subsystem can be partitioned into subsystem elements, pre-designed in agreement with IEC 61508, with different systematic capability level. Then the systematic capability of one subsystem element can potentially limit the SIL of its subsystem. For additional details, see IEC 61508-2.

7.3.2.2 Requirements for the avoidance of systematic failures

The following measures shall all be applied if applicable:

– appropriate selection, combination, arrangements, assembly and installation of components, including cabling, wiring and any interconnections:
apply manufacturer's application notes, e.g. user manual, installation instructions, specifications and use of good engineering practice (e.g. IEC 60204-1);

– use of the subsystem and subsystem elements within the manufacturer’s specification and installation instructions;

– compatibility: use components with compatible operating characteristics;

– withstanding specified environmental conditions:
design the subsystem so that it is capable of working in all expected environments and in any foreseeable adverse conditions (within the defined limit of use), for example temperature, humidity, vibration and electromagnetic fields;

– use of components that are in accordance with an applicable standard and have their failure modes well-defined: to reduce the risk of undetected faults by the use of components with specific characteristics;

NOTE 1 Components such as hydraulic or pneumatic valves can require cyclic switching to avoid the failure mode of non-switching or unacceptable increase in switching times. In this case, a periodic test can be necessary.

– use of suitable materials and adequate manufacturing:
selection of material, manufacturing methods and treatment in relation to, for example stress, durability, elasticity, friction, wear, corrosion, temperature, conductivity, dielectric strength;

– correct dimensioning and shaping:
consider the effects of, for example, stress, strain, fatigue, temperature, surface roughness, manufacturing tolerances.

NOTE 2 IEC 61508-2:2010, Annex F specifies techniques and measures for avoidance of systematic failures during design and development of application-specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), programmable logic devices (PLDs), etc.

NOTE 3 Table B.1 to B.5 of IEC 61508-2:2010, Annex B give techniques and measures to avoid failures in safety-related systems which can be useful during specification, design, integration, operation, maintenance and validation phases.

NOTE 4 Annexes A to D of ISO 13849-2:2012 provide principles for mechanical, pneumatic, hydraulic and electrical systems.

In addition, one or more of the following measures shall be applied if applicable:

a) hardware design review (e.g. by inspection or walk-through):
to reveal by reviews and/or analysis discrepancies between the specification and implementation;

NOTE 5 In order to reveal discrepancies between the specification and implementation, any points of doubt or potential weak points concerning the realization, the implementation and the use of the product are documented so they can be resolved; in an inspection procedure the author is passive and the person inspecting is active whilst on a walk-through procedure the author is active and the person inspecting is passive.

b) computer-aided design tools capable of simulation or analysis:
perform the design procedure systematically and include appropriate automatic construction elements that are already available and tested;

NOTE 6 These tools can be qualified by specific testing, or by an extensive history of satisfactory use, or by independent verification of their output for the particular subsystem that is being designed.

c) simulation:
perform a systematic simulation of a subsystem design in terms of both the functional performance and the correct dimensioning of their components.

NOTE 7 The function of the subsystem can be simulated on a computer via a software behavioral model where individual components of the circuit each have their own simulated behaviour, and the response of the subsystem in which they are connected is examined by looking at the marginal data of each component.

7.3.2.3 Requirements for the control of systematic failures

The following measures shall all be applied if applicable:

a) measures to control the effects of insulation breakdown, voltage variations and interruptions, overvoltage and undervoltage: subsystem behaviour in response to insulation breakdown, voltage variations and interruptions, overvoltage and undervoltage conditions shall be pre-determined so that the subsystem can achieve or maintain a safe state;

NOTE 1 Further information can be found in IEC 60204-1 and IEC 61508-7:2010, Clause A.8.

b) measures to control or avoid the effects of the physical environment (for example, temperature, humidity, water, vibration, dust, corrosive substances, electromagnetic interference and its effects): subsystem behaviour in response to the effects of the physical environment shall be pre-determined so that the SCS can achieve or maintain a safe state. See also e.g. IEC 60529, IEC 60204-1 and IEC 60721 (all parts);

c) measures to control or avoid the effects of temperature increase or decrease, if temperature variations can occur: the subsystem should be designed so that, for example, over-temperature can be detected before it begins to operate outside specification;

NOTE 2 Further information can be found in IEC 61508-7:2010, Clause A.10.

d) measures to control the effects of hose breakdown, pressure variations and interruptions, too low or too high pressure: subsystem behaviour in response to hose breakdown, pressure variations and interruptions, too low or too high pressure shall be pre-determined so that the subsystem can achieve or maintain a safe state.

NOTE 3 Further information can be found in ISO 4414:2010 for pneumatic systems or ISO 4413 for hydraulic systems.

When PELV/SELV power supply (see IEC 60364-4-41) is used, the over voltage at the output in event of a single fault shall be taken into account in the analysis of the effects of over voltage including the possibility of common cause failure.

NOTE 4 Over voltage ranges are given for example in IEC 60950-1, IEC 61204-7, IEC 62477 (all parts), IEC 60449.

In addition, the following basic safety principles, as appropriate, shall be applied for the control of systematic failures:

– use of de-energization:
the subsystem should be designed so that with loss of its power supply, a safe state can be achieved or maintained;

NOTE 5 For further information, see ISO 13849-2.

– measures for controlling the effects of errors and other effects arising from any data communication process (see IEC 61508-2:2010, 7.4.11).

Depending on the selected architecture of the subsystem, the following well tried safety principles, as appropriate, shall be applied to the subsystem element for the control of systematic failures:

– failure detection by automatic tests;
– tests by comparison of redundant hardware;

NOTE 6 For further information, see ISO 12100:2010, 6.2.12.4.

– diverse hardware;
– operation in the positive mode (e.g. a limit switch is pushed when a guard is opened);
– mechanically linked contacts;
– direct opening action;
– oriented mode of failure;

NOTE 7 For further information, see ISO 12100:2010, 6.2.12.3.

– over-dimensioning by a suitable factor can improve reliability and an appropriate factor of over-dimensioning shall be determined.

NOTE 8 For further information, see ISO 13849-2 and Annex A of IEC 61508-2:2010.

7.3.2.4 Electromagnetic immunity

Subsystem design shall take into account the requirements of 6.6.

7.3.2.5 Security aspects

Subsystem design shall take into account the requirements of 6.8.

7.3.3 Fault consideration and fault exclusion

7.3.3.1 General

All subsystem elements shall be designed to achieve the required safety requirement specification. The ability to resist faults shall be assessed. Where not explicitly stated otherwise, the requirements of this Clause 7 apply independently of the required safety integrity of the safety function.

7.3.3.2 Fault consideration

To estimate the capability of a subsystem element to reach a certain safe state, an analysis of each subsystem element shall be performed to determine all relevant faults and their corresponding failure modes. Whether a failure is a safe or a dangerous failure depends on the SCS and the intended safety functions, including fault reaction function.

Analysis technique such as failure mode and effect analysis (FMEA, see IEC 60812), fault tree analysis (FTA, see IEC 61025) or event tree analysis (ETA, see IEC 62502) can be carried out to establish the faults that are to be considered for those components.

The probability of each failure mode shall be determined based on the probability of the associated fault(s) taking into account the intended use and can be derived from sources such as:

– dependable failure rate data collected from field experience by the manufacturer and relevant to the intended use;
– component failure data from a recognised industry source and relevant to the intended use;
– failure mode data;
– failure rate data derived from the results of testing and analysis.

In general, 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 (known as a dependent 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.

7.3.3.3 Fault exclusion

It is not always possible to evaluate subsystems without assuming that certain faults may be excluded. Fault exclusion is a compromise between technical safety requirements and the theoretical possibility of occurrence of a fault.

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.

Fault exclusion is only applicable for certain failures of an element and it is up to the designer (manufacturer or integrator) to prove the exclusion of the respective faults based on the limits set forward by the design and use. Such fault exclusion is only possible provided that the technical improbability of them occurring can be justified based on the known laws of physical science. Any such fault exclusions shall be justified and documented.

The application of fault exclusion to certain faults for an element inside a subsystem does not limit the necessity of the application of systematic measures.

It is possible some faults are excluded by the manufacturer and some by the subsystem integrator.

Fault exclusion is one principle to limit the failure of a component/subsystem; also other methods are possible (e.g. architectures, limitation of systematic failures).

There shall be a specific characterization of the type of fault that is excluded. It would not be acceptable to state simply that a component will not break, distort or degrade due to wear. It would be necessary to state the direct influence under which the component will not break, distort or degrade due to wear. E.g. the component will have no faults when subjected to a force of X Newtons from direction Y.

The fault exclusion shall be justifiable under all expected industrial environments including temperature, pressure, vibration, pollution, corrosive atmosphere, etc.

NOTE 1 Useful information on fault exclusions is available in ISO 13849-2:2012, Annex A to D.

Fault exclusion can only be applied for the entire subsystem when all dangerous failures of a subsystem can be excluded.

LIMITATION: For some applications, it is not expected that all failures can be excluded with sufficient confidence for SIL 3. The following non exhaustive list provides an indication of (non-predesigned) subsystems with a hardware fault tolerance of zero and where fault exclusions have been applied to faults that could lead to a dangerous failure where a maximum of SIL 2 can be appropriate provided that sufficient justification is given:

– position switch with mechanical aspects with HFT of 0;
– leakage of a fluid power valve (where leakage is dangerous failure).

NOTE 2 This limitation does not apply to pre-designed subsystems used within their specification.

7.3.3.4 Functional testing to detect fault accumulation and undetected faults

In a redundant system, an accumulation of faults over time might lead to a loss of the safety function. In a single channel system, undetected faults might also lead to a loss of the safety function.

For an SCS with non-electronic technology and using automatic monitoring to achieve the necessary diagnostic coverage for the required safety performance, the monitoring function cannot be possible unless there is a change of state, e.g. at every operating cycle. If, in such a case, there is only infrequent operation, the probability of occurrence of an undetected fault is increased. When a functional test is necessary to detect a possible accumulation of faults or a undetected fault before the next demand, it shall be made within the following test intervals:

– at least every month for SIL 3;
– at least every 12 months for SIL 2.

EXAMPLE: The control system of a machine can demand these tests at the required intervals e.g. by visual display unit or signal lamp, and can monitor the tests and stop the machine if the test is omitted or fails.

7.3.4 Failure rate of subsystem element

7.3.4.1 General

The mathematical probability of failure of a subsystem element can be characterized by one of three parameters: λ (Lambda), $MTTF$ (Mean Time To Failure) or B_{10}.

NOTE Although the parameters above can be delivered in several valuable formats, the typical formats are:

– λ: failures per hour;

– $MTTF$: mean time to failures expressed in years;

– B_{10}: switching cycles of wearing components.

For the estimation of the parameters of a subsystem element, the hierarchical procedure for finding data shall be, in the order given:

a) use manufacturer's data;

b) use Annex C of this document;

c) choose a $MTTF_D$ of ten years.

The data could be delivered as values with respect to the dangerous failures (λ_D, $MTTF_D$, B_{10D}) or with respect to all failures (λ, $MTTF$, B_{10}).

To determine the dangerous failures from the overall failures, the different failure modes of the subsystem element should be taken into account. It is typically assumed that not all failures modes lead to a dangerous failure. This depends mainly on the application, so generally the failure mode data used should reflect practical application of the components. A precise way of determining the "failure modes" of a subsystem element is to carry out an FMEA. If no specific or sufficient knowledge and information is available concerning the failure modes, 50 % of the failures can be estimated as dangerous.

7.3.4.2 Relationship of relevant parameters

For subsystem elements, constant failure rates (λ) of the subsystem elements are assumed. The following basic equations can be used:

$$\lambda = \frac{1}{MTTF} \tag{3}$$

$$\lambda_D = \frac{1}{MTTF_D} \tag{4}$$

NOTE 1 For calculation purposes, $MTTF$ can be assumed equal to mean operating time between failures ($MTBF$).

$MTTF$ and $MTTF_D$ are mostly indicated in years [a]. λ values are commonly indicated in FIT (FIT = Failure In Time) where 1 FIT means one failure in 10^9 hours.

$$1\,FIT = 1 \times 10^{-9}\,\mathrm{h}^{-1} \tag{5}$$

One year is approximately 8 760 hours. Therefore, a $MTTF$ value can be converted into a λ value.

$$\lambda = \frac{1}{MTTF \times 8760\frac{\mathrm{h}}{\mathrm{a}}} \tag{6}$$

NOTE 2 Example, $MTTF$ = 1 000 a:

$$\lambda_{example} = \frac{1}{1\,000\text{a} \times 8\,760 \frac{\text{h}}{\text{a}}}$$

$$\lambda_{example} = \frac{1}{8\ 760\ 000\text{h}}$$

$$\lambda_{example} = \frac{1}{8\ 760\ 000}\text{h}^{-1}$$

$$\lambda_{example} = 114{,}155 \times 10^{-9}\text{h}^{-1}$$

$$\lambda_{example} = 114{,}155\ FIT$$

For pneumatic, mechanical and electromechanical components (pneumatic valves, relays, contactors, position switches, cams of position switches, etc.) it can be difficult to calculate the mean time to dangerous failure ($MTTF_D$ for components), which is given in years. Usually the manufacturers of these kinds of components only give the mean number of cycles until 10 % of the components fail dangerously (B_{10D}). This Clause 7 gives a method for calculating an $MTTF_D$ for components by using B_{10D} given by the manufacturer related closely to the application dependent cycles.

NOTE 3 Hydraulic components are mostly characterized with $MTTF_D$.

If the appropriate basic and well-tried safety principles are met, the $MTTF_D$ value for a single pneumatic, electromechanical or mechanical component can be estimated.

The mean number of cycles until 10 % of the components fail dangerously (B_{10D}) should be determined by the manufacturer of the component in accordance with relevant product standards for the test methods (e.g. IEC 60947-5-1, ISO 19973, IEC 61810). The dangerous failure modes of the component have to be defined, e.g. sticking at an end position or change of switching times. If not all the components fail dangerously during the tests (e.g. seven components tested, only five fail dangerously), an analysis taking into account the components that were not dangerously failed components should be performed.

With B_{10D} and n_{op}, the mean number of annual operations, $MTTF_D$ for components can be calculated as

$$MTTF_D = \frac{B_{10D}}{0{,}1\ \ n_{op}} \tag{7}$$

where

$$n_{op} = \frac{d_{op} \times h_{op} \times 3600 \frac{\text{s}}{\text{h}}}{t_{cycle}} \tag{8}$$

and with the following assumptions having been made on the application of the component:

h_{op} is the mean operation, in hours per day;

d_{op} is the mean operation, in days per year;

t_{cycle} is the mean time between the beginning of two successive cycles of the component. (e.g. switching of a valve) in seconds per cycle.

In terms of failure rate λ, the following relationship can be expressed as

$$\lambda_D = \frac{0{,}1\ C}{B_{10D}} = \frac{0{,}1\ n_{op}}{B_{10D} \times 8\ 760 \frac{h}{a}} \quad (9)$$

where C (C = n_{op} / 8 760) is the duty cycle or mean operation per hour.

The relation between B_{10D}, B_{10} and the ratio of dangerous failure (RDF) is

$$B_{10D} = \frac{B_{10}}{\text{ratio of dangerous failure}} \quad (10)$$

The useful lifetime of the component is limited to T_{10D}, the mean time until 10 % of the components fail dangerously:

$$T_{10D} = \frac{B_{10D}}{n_{op}} \quad (11)$$

NOTE 4 For electronic systems, the exponential distribution is applicable. For non-electronic systems, the exponential distribution is not applicable. The Weibull distribution (see also IEC 61649) is more appropriate, but parameters and calculations are difficult to apply. However, when using exponential distribution for non-electronic components within the limits of T_{10D} then the results of the calculations are pessimistic and the formula with $1\text{-}e^{-\lambda t}$ could be applied as a simplified method.

If the ratio of dangerous failure is estimated less than 0,5 (50 % dangerous failure) the useful lifetime of the component is limited to twice T_{10}.

NOTE 5 Similar to Formula (11), T_{10} is evaluated by $T_{10} = \frac{B_{10}}{n_{op}}$.

For further details, see IEC TS 63394:2023, Clause H.6.

The ratio of dangerous failure is estimated as 0,5 (50 % dangerous failure) if no other information (e.g. product standard) is available.

7.4 Architectural constraints of a subsystem

7.4.1 General

In the context of hardware safety integrity, the highest safety integrity level that can be claimed for an SCS is limited by the hardware fault tolerances (HFT) and safe failure fractions (SFF) of the subsystems that carry out that safety function. Table 6 specifies the highest safety integrity level that can be claimed for an SCS that uses a subsystem taking into account the hardware fault tolerance and safe failure fraction of that subsystem. The architectural constraints given in Table 6 shall be applied to each subsystem developed according to Clause 7. With respect to these architectural constraints:

a) a hardware fault tolerance of N means that N+1 faults could cause a loss of the safety function. In determining the hardware fault tolerance, no account is taken of other measures that can control the effects of faults such as diagnostics; and

b) where one fault directly leads to the occurrence of one or more subsequent faults, these shall be considered as a single fault;

c) in determining hardware fault tolerance, certain faults may be excluded, provided that the likelihood of them occurring is very low in relation to the safety integrity requirements of the subsystem, see 7.3.3.3.

A subsystem that comprises only a single subsystem element shall satisfy the requirements of Table 4. In particular, for an HFT 0 (zero fault tolerance) subsystem element of SIL 3, a SFF of greater than 99 % shall be achieved by an SCS diagnostic function.

When two or more pre-designed subsystems are combined into one redundant subsystem, the architectural constraints of the combined subsystem can be determined. This can be done by taking the subsystem with the highest SIL according to the architectural constraints and looking for the corresponding SIL in Table 6 in column HFT 0. This will return the applicable SFF range. The SIL of the combined subsystem shall be derived by increasing the HFT by one in the same SFF range according to IEC 61508-2:2010, 7.4.4.2.4

NOTE 2 This procedure is only applicable for combining subsystems with a defined SIL.

Table 6 – Architectural constraints on a subsystem: maximum SIL that can be claimed for an SCS using the subsystem

Safe failure fraction (SFF)	Hardware fault tolerance (HFT) (see NOTE 1)		
	0	1	2
< 60 %	Not allowed (for exceptions see NOTE 3)	SIL 1	SIL 2
60 % to < 90 %	SIL 1	SIL 2	SIL 3
90 % to < 99 %	SIL 2	SIL 3	SIL 3 (see NOTE 2)
≥ 99 %	SIL 3	SIL 3 (see NOTE 2)	SIL 3 (see NOTE 2)

NOTE 1 A hardware fault tolerance of N means that N+1 faults could cause a loss of the safety function.

NOTE 2 SIL 4 is not considered in this document. For SIL 4, see IEC 61508-1.

NOTE 3 See ~~7.4.3.2~~ 7.5.3, where subsystems which have a safe failure fraction of less than 60 % and zero hardware fault tolerance that use well-tried components can be considered to achieve SIL 1; or for subsystems where fault exclusions have been applied to faults that could lead to a dangerous failure.

NOTE 4 In IEC 62061:2015 the maximum SIL that could be claimed was named SILCL.

NOTE 5 See 7.3.3.3 for limitation of SIL when applying fault exclusion.

NOTE 6 For HFT 0 at $SFF \geq 99$ %, it is only possible when there is continuous monitoring of the correct functioning of the element. Typically, electronic technology will be required to achieve this.

7.4.2 Estimation of safe failure fraction (SFF)

To estimate the SFF, an analysis (e.g. fault tree analysis, failure mode and effects analysis) of each subsystem shall be performed to determine all relevant faults and their corresponding failure modes. Whether a failure is a safe or a dangerous failure depends on the SCS and the intended safety function, including fault reaction function (see 7.4.3). The probability of each failure mode shall be determined based on the probability of the associated fault(s) taking into account the intended use and may be derived from sources such as:

a) dependable failure rate data collected from field experience by the manufacturer and relevant to the intended use;

b) ~~component~~ failure rate data from a recognised industry source and relevant to the intended use;

c) failure rate data derived from the results of testing and analysis.

NOTE 1 Information of the failure rates for electrical/electronic component can be found in several sources including: MIL-HDBK 217 F, MIL-HDBK 217 F (Appendix A), SN 29500 Parts 7 and 11, IEC 61709, FMD-2016, OREDA Handbook, EXIDA Safety Equipment Reliability Handbook and EXIDA Electrical & Mechanical Component Reliability Handbook.

NOTE 2 Failure rate data can be provided by manufacturers.

NOTE 3 Some component standards provide relevant data (e.g. Annex K of IEC 60947-4-1:2018).

NOTE 4 Lists of faults to be considered for mechanical, pneumatic, hydraulic and electrical technologies are given in Annexes A, B, C and D of ISO 13849-2:2012.

In general, the SFF can be calculated as follows:

$$SFF = \frac{\sum \lambda_S + \sum \lambda_{DD}}{\sum \lambda_S + \sum \lambda_D} \tag{12}$$

where

λ_S is the rate of safe failure,

$\sum \lambda_S + \sum \lambda_D$ is the overall failure rate,

λ_{DD} is the rate of dangerous failure which is detected by the diagnostic functions,

λ_D is the rate of dangerous failure.

The failure of an element that plays a part in implementing the safety function but has no direct (adverse) effect on the safety function is termed a no effect failure and is not considered as a safe failure (λ_S). Therefore, it shall not be used for SFF calculations.

For non-electronic components, λ_S is typically assumed as equal to 0 or is negligible, because in most cases it is insignificant in comparison to λ_D. In this case, the following simplification can be applied (see also example in Clause B.4):

~~$$SFF = \frac{\sum \lambda_S + \sum \lambda_{DD}}{\sum \lambda_S + \sum \lambda_D} = \frac{\sum \lambda_{DD}}{\sum \lambda_D}$$~~

$$SFF = \frac{\sum \lambda_S + \sum \lambda_{DD}}{\sum \lambda_S + \sum \lambda_D} \approx \frac{\sum \lambda_{DD}}{\sum \lambda_D} \tag{13}$$

EXAMPLE ~~2~~1 Where the hardware fault tolerance of a subsystem is equal to 0, the SFF becomes

~~$$SFF = \frac{\lambda_{DD1}}{\lambda_{D1}} = \frac{DC_1 \lambda_{D1}}{\lambda_{D1}} = DC_1$$~~

$$SFF \approx \frac{\lambda_{DD1}}{\lambda_{D1}} = \frac{DC_1 \lambda_{D1}}{\lambda_{D1}} = DC_1$$

where DC_1 is the diagnostic coverage of subsystem element 1.

EXAMPLE ~~3~~2 Where the hardware fault tolerance of a subsystem is equal to 1, SFF becomes

$$SFF = \frac{\lambda_{DD1} + \lambda_{DD2}}{\lambda_{D1} + \lambda_{D2}} = \frac{DC_1\lambda_{D1} + DC_2\lambda_{D2}}{\lambda_{D1} + \lambda_{D2}} = \frac{\frac{DC_1}{MTTF_{D1}} + \frac{DC_1}{MTTF_{D2}}}{\frac{1}{MTTF_{D1}} + \frac{1}{MTTF_{D2}}}$$

$$SFF \approx \frac{\lambda_{DD1} + \lambda_{DD2}}{\lambda_{D1} + \lambda_{D2}} = \frac{DC_1\lambda_{D1} + DC_2\lambda_{D2}}{\lambda_{D1} + \lambda_{D2}} = \frac{\frac{DC_1}{MTTF_{D1}} + \frac{DC_2}{MTTF_{D2}}}{\frac{1}{MTTF_{D1}} + \frac{1}{MTTF_{D2}}}$$

where DC_1 and DC_2 are the diagnostic coverages respectively of subsystem element 1 and 2 (see also 7.4.2 for relationship between λ and $MTTF$).

7.4.3 Behaviour (of the SCS) on detection of a fault in a subsystem

7.4.3.1 General

The detection of a dangerous fault in any subsystem that has a hardware fault tolerance of more than zero shall result in the performance of the specified fault reaction function.

The specification can allow isolation of the faulty part of the subsystem to continue safe operation of the machine while the faulty part is repaired. In this case, if the faulty part is not repaired within the estimated maximum time, as assumed in the calculation of the PFH, then a second fault reaction shall be performed to achieve a safe state.

Where the SCS is designed for online repair, isolation of a faulty part shall only be applicable where this does not increase the PFH of the SCS above that specified in the SRS.

As long as operation is continued and hardware fault tolerance is reduced to zero, the requirements of 7.4.3.2 apply.

7.4.3.2 Fault reaction function

Where a diagnostic function is necessary to achieve the required PFH or safe failure fraction and the subsystem has a hardware fault tolerance of zero, then

– the sum of the diagnostic test interval and the time to perform the specified fault reaction function to achieve or maintain a safe state shall be shorter than the process safety time (e.g. see ISO 13855); or,

– when operating in high demand mode of operation, the ratio of the diagnostic test rate to the demand rate shall equal or exceed 100.

Where performance of a fault reaction function as part of an SCS that is specified as SIL 3 has resulted in the machine being stopped, subsequent normal operation of the machine via the SCS (e.g. enabling re-start of the machine) shall not be possible until the fault has been repaired or rectified. For an SCS with a specified safety performance of less than SIL 3, the behavior of the machine after performance of a fault reaction function (e.g. re-starting normal operation) shall depend on the specification of relevant fault reaction functions (see 5.2.2).

7.4.3.3 Diagnostic coverage (DC)

Diagnostic coverage (DC) can be calculated as the fraction of dangerous failures by using the following equation:

~~$DC = \sum \lambda_{DD} / \sum \lambda_D$~~

$$DC = \frac{\sum \lambda_{DD}}{\sum \lambda_D} \quad (14)$$

where λ_{DD} is the rate of detected dangerous hardware failures and λ_D is the rate of dangerous hardware failures.

For the estimation of DC, in most cases, failure mode and effects analysis (FMEA – see IEC 60812), failure mode effects and diagnostic analysis (FMEDA) or equivalent methods can be used. In this case, all relevant faults and/or failure modes should be considered.

For a simplified approach to estimating DC, see Annex D.

NOTE Annex C of IEC 61508-2:2010 provides further information.

7.4.4 Realization of diagnostic functions

Each subsystem shall be provided with associated diagnostic functions that are necessary to fulfil the requirements for architectural constraints and the PFH.

The diagnostic functions are considered as separate functions that can have a different structure than the safety function and can be performed by

– the same subsystem which requires diagnostics; or
– other subsystems of the SCS; or
– subsystems of the SCS not performing the safety function.

Diagnostic functions shall satisfy the following:

– applicable requirements for the avoidance of systematic failure; and
– applicable requirements for the control of systematic failure.

NOTE 1 Timing constraints applicable to the testing of the subsystem performing a diagnostic function can differ from those applicable to safety functions.

NOTE 2 The necessity of checking the diagnostic function can depend on aspects such as the safety integrity level, the demand rate, the technology used and application specific capabilities.

A clear description of the SCS diagnostic function(s), their failure detection/reaction, and an analysis of their contribution towards the safety integrity of the associated safety functions shall be provided.

To apply the simplified approach of this document for the estimation of PFH of subsystems, the following shall apply:

SCS diagnostic function(s) shall as a minimum be implemented so that the PFH and the systematic safety integrity are the same as those specified for the corresponding safety function(s),

or

where the PFH is of an order of magnitude greater than that specified for the safety function, then a test shall be performed to determine whether diagnostic function(s) remain operational; a test of the diagnostic function(s) shall be carried out at a minimum of 10 times at equal intervals during the proof test interval for the subsystem.

NOTE 3 Architectural constraints on hardware safety integrity do not apply to the realization of diagnostic function(s).

NOTE 4 A test of the diagnostic function(s) is foreseen to cover, as far as practicable, 100 % of those parts implementing the diagnostic function(s).

NOTE 5 Where a diagnostic function is implemented by the logic solver of the SCS, it can be unnecessary to perform a separate test of the diagnostic function as its failure can be revealed as a failure of the safety function.

NOTE 6 A test can be performed by either external means (e.g. test equipment) or internal dynamic checks (e.g., embedded within the logic solver) of the SCS.

7.5 Subsystem design architectures

7.5.1 General

The architecture of any subsystem described in this subclause can be used to evaluate the architectural constraints and to estimate the PFH; see Annex H.

NOTE The figures in 7.5 represent a logical view of the subsystem architectures and are not intended to represent any specific physical connection schemes. A hardware fault tolerance of 1 is represented by parallel subsystem elements but the physical connections will depend on the application of the subsystem.

7.5.2 Basic subsystem architectures

7.5.2.1 Basic subsystem architecture A: single channel without a diagnostic function

In this architecture (see Figure 8), any dangerous failure of a subsystem element causes a failure of the safety function. This architecture corresponds to a hardware fault tolerance of 0.

In high or continuous mode of operation, architecture A shall not rely on a proof test.

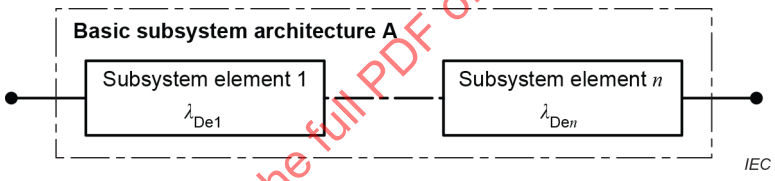


Figure 8 – Basic subsystem architecture A logical representation

7.5.2.2 Basic subsystem architecture B: dual channel without a diagnostic function

This architecture (Figure 9) is such that a single failure of any subsystem element does not cause a loss of the safety function. This architecture corresponds to a hardware fault tolerance of 1.

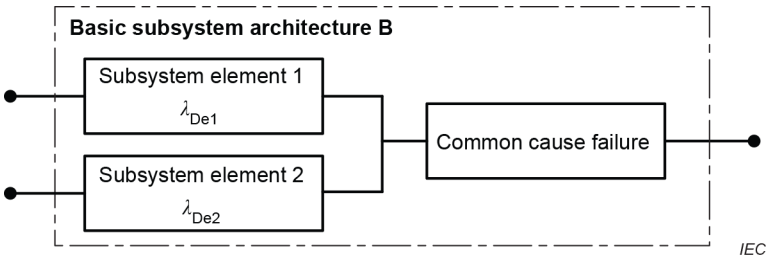


Figure 9 – Basic subsystem architecture B logical representation

7.5.2.3 Basic subsystem architecture C: single channel with a diagnostic function

In this architecture (see Figure 10), any undetected dangerous fault of the subsystem element leads to a dangerous failure of the safety function.

Where a fault of a subsystem element is detected, the diagnostic function(s) initiates a fault reaction function (see 7.4.3). This architecture corresponds to a hardware fault tolerance of 0.

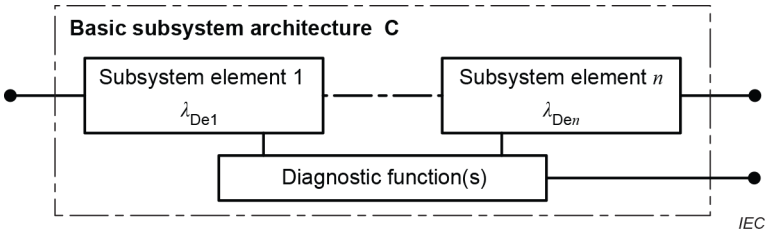


Figure 10 – Basic subsystem architecture C logical representation

7.5.2.4 Basic subsystem architecture D: dual channel with a diagnostic function(s)

This architecture (see Figure 11) is such that a single failure of any subsystem element does not cause a loss of the safety function. Where a fault of a subsystem element is detected, the diagnostic function(s) initiates a fault reaction function (see 7.4.3). This architecture corresponds to a hardware fault tolerance of 1.

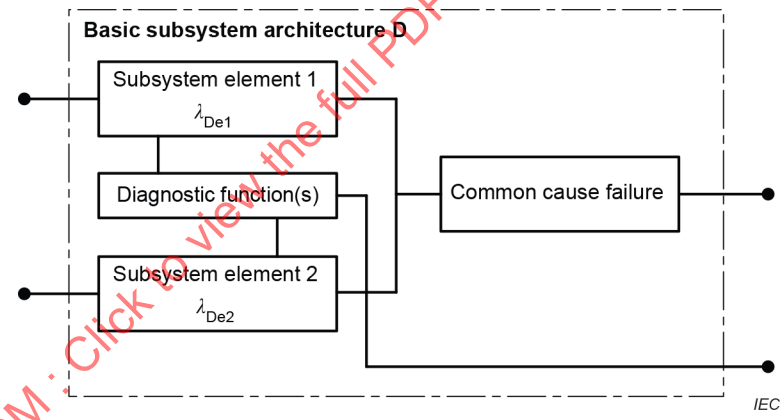


Figure 11 – Basic subsystem architecture D logical representation

7.5.3 Basic requirements

As shown in Table 7, the basic requirements depending on the architectural constraints and the basic subsystem architectures shall be applied.

Table 7 – Overview of basic requirements and interrelation to basic subsystem architectures

Basic requirements	Hardware fault tolerance (HFT)				Comments / Examples
	0		1		
	SFF		*SFF*		
	<60 %	≥ 60 %	<60 %	≥ 60 %	
Basic safety principles	M	M	M	M	Use of suitable materials ISO 13849-2:2012, Annex A to D
Well-tried safety principles	M	M	M	M	Mechanically linked contacts and contacts with direct opening action ISO 13849-2:2012, Annex A to D
Well-tried components	M	--	--	--	Contactor (IEC 60947-4-1) ISO 13849-2:2012, Annex A to D
CCF	not relevant	M	M	M	
Type of basic subsystem architecture	**A**	**C**	**B**	**D**	
M = mandatory; -- = no requirement NOTE Table 6 for architectural constraints is still applicable.					

7.6 *PFH* of subsystems

7.6.1 General

The following parameters have to be determined to be able to determine the PFH:

– subsystem architecture (see 7.5);
– DC and test intervals (see 7.4.3 and 7.4.4);
– CCF (see Annex E);
– λ_D or $MTTF_D$ of subsystem elements (see 7.3.4);
– useful lifetime.

NOTE Because a typical machine life is about 20 years, a useful lifetime of 20 years is preferred. The intention is to clarify the maximum usage period for the subsystem. For components with wear out characteristics, useful lifetime can be limited by T_{10D}.

7.6.2 Methods to estimate the *PFH* of a subsystem

One of the following methods of Annex H may be used to calculate the PFH as simplified approach:

– allocation table approach (see Clause H.1);
– formulas (see Clause H.2).

Modelling based on e.g. fault tree analysis (see B.6.6.5 of IEC 61508-7:2010 and IEC 61025), Markov models (see B.6.6.6, C.6.4 of IEC 61508-7:2010 and IEC 61165) or reliability block diagrams (see C.6.4 of IEC 61508-7:2010) is always possible.

7.6.3 Simplified approach to estimation of contribution of common cause failure (CCF)

Knowledge of the susceptibility of a subsystem to CCF is required to contribute to the estimation of the PFH of a subsystem.

The probability of occurrence of the CCF will usually be dependent upon a combination of technology, architecture, application and environment. The use of Annex E will be effective in avoiding many types of CCF.

8 Software

8.1 General

All lifecycle activities of safety-related application software shall focus on the avoidance of faults introduced during the software lifecycle. The main objective of the following requirements is to produce readable, understandable, testable, maintainable and correct software.

Where the software performs both non-safety and safety functions, then all of the software shall be treated as safety-related, unless sufficient independence between the functions can be demonstrated in the design. It is therefore preferable to separate non-safety functions such as basic machine functions from safety functions wherever practicable.

This document shall only be used for application software that is running in a pre-designed platform according to IEC 61508 or other functional safety standards linked to IEC 61508 e.g. IEC 61131-6.

NOTE In the remainder of this clause, application software is also referred to as software.

8.2 Definition of software levels

This document describes three different levels of application software, see Table 8.

Table 8 – Different levels of application software

SW level	Main principle	Subprinciple	Example
1	Platform (combination of hardware and software) pre-designed according to IEC 61508, or other functional safety standards linked to IEC 61508 e.g. IEC 61131-6. Application software making use of a limited variability language (LVL).	Application software complying with this document.	Safety PLC with LVL or Safety programmable relay
2	Platform (combination of hardware and software) pre-designed according to IEC 61508, or other functional safety standards linked to IEC 61508 e.g. IEC 61131-6. Application software making use of another language than a limited variability language (LVL).	Application software complying with this document.	Safety PLC with FVL (FVL complying with this document.)
3		Application software complying with IEC 61508-3.	Safety PLC with LVL or FVL (FVL according IEC 61508)

NOTE 1 Software Level 2 is introduced to support Full Variability Language, but limited to SIL 2. For SIL 3 compliant application SW, so-called software level 3, follow IEC 61508-3.

NOTE 2 For other types of platforms no requirements are set forward in this document. IEC 61508-2 and 61508-3 describe how to handle such systems.

The programming language (instruction set) to be used for the application software has to be in the safety-related scope of the platform, pre-designed according to IEC 61508, or other functional safety standards linked to IEC 61508 e.g. IEC 61131-6.

The programming language to be used and the tools (of the software development lifecycle) shall be suitable for the creation of safety related application software on the platform; see 8.4.1.3.

In this context, the platform described in Table 8 shall require only the application software to execute its safety related functionality.

NOTE For example, elements such as systems on chip or microcontroller boards are not platforms in this sense.

Software level 3 is not further described in this document since it is covered by correct application of the respective parts of IEC 61508. A high level of competence is required in order to design according to SW level 2 or 3. Factors that make the use of IEC 61508-3 (software level 3) more appropriate than the use of this document (software level 2) are:

– high degree of complexity of the safety function(s);
– large number of safety functions;
– large project size.

The software safety lifecycle requirements for the different SW levels are detailed in the following subclauses:

– SW level 1: see 8.3;
– SW level 2: see 8.4.

8.3 Software – Level 1

8.3.1 Software safety lifecycle – SW level 1

8.3.1.1 Maximum achievable SIL – SW level 1

The maximum achievable SIL for SW level 1 is SIL 3.

8.3.1.2 Software safety lifecycle model – SW level 1

A software safety lifecycle model which is resolved into distinct phases shall be used (e.g. V-model), including management and documentation activities to achieve the required level of safety.

Any software lifecycle model may be used provided all the objectives and requirements of this Subclause 8.3 are met. Safety-related software shall be validated as described in 9.5.4.

SW level 1 is of reduced complexity due to the use of pre-designed safety-related hardware and software modules. Therefore, the simplified V-model in Figure 12 is applicable. The design of customized, or self-created software modules can be necessary (e.g. in the case of the library modules provided by the component manufacturer being inadequate or not suitable). The design of software modules customized by the designer is an additional activity which shall be carried out according to the V-model in Figure 13.

NOTE 1 A software module (or briefly module) is a functional unit of the software, which is typically only accessible through its input and output interface. It is reusable and facilitates the modular software development. Software modules are often part of a library. In PLC-programming, software modules are functions or function blocks.

NOTE 2 The V-model is a static model used to structure the software design into small parts. It does not introduce any sequence of creation of specifications or implementation. The left side represents requirements; i.e. things to achieve. The right side details testing of the software.

NOTE 3 On the left side of the V-model, the output of each phase is reviewed. Review means to check the output of a phase in the V-model against the requirements of the input of the same phase. The arrow 'Review' represents the first step of the software verification. Further information on the level of independence of review and testing/verification is available in Annex J.

NOTE 4 The lifecycle is accompanied by project management techniques and processes appropriate for the size and scope of the project.

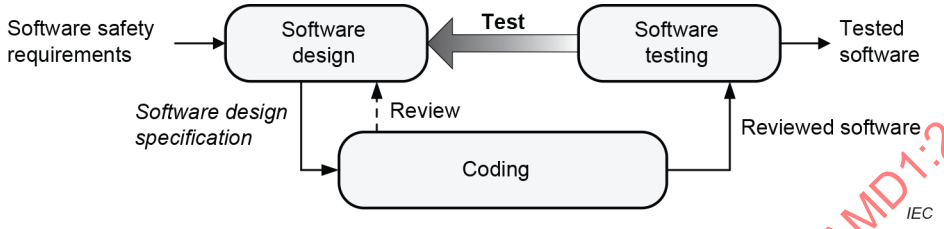


Figure 12 – V-model for SW level 1

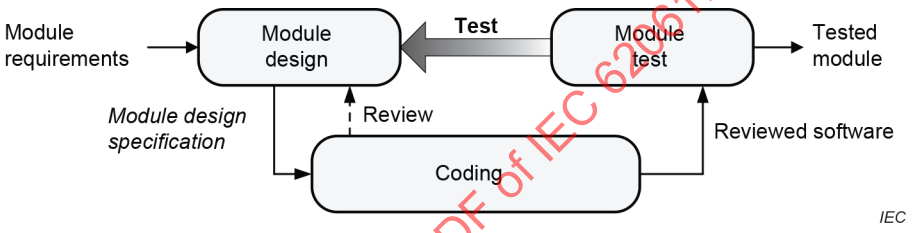


Figure 13 – V-model for software modules customized by the designer for SW level 1

NOTE 5 In the V-models the arrow 'Test' represents the results of test cases according to the specification and, in addition, the need for more precise test case requirements and specifications.

NOTE 6 The result of Figure 13 is an input to the coding of Figure 12.

8.3.1.3 Tools usage – SW level 1

The tools shall be applied according to the instructions of the relevant manufacturer of the safety-related system(s) (e.g. PLC, electro sensitive protective equipment).

8.3.2 Software design – SW level 1

8.3.2.1 General – SW level 1

Where software is to be used in any part of an SCS implementing a Safety Function, a software safety requirements specification shall be developed, documented and managed throughout the lifecycle of the SCS.

The software safety requirements specification shall be developed for each subsystem on the basis of the SCS specification and architecture.

8.3.2.2 Software safety requirements – SW level 1

To support the software design process the following information shall be considered:

a) specification of the safety function(s) (see 5.2);

b) configuration or architecture of the SCS (e.g. hardware architecture, wiring diagram, safety-related inputs and outputs);

c) response time requirements;

d) operator interfaces and controls, such as: switches, joysticks, mode selector, dials, touch sensitive control devices, keypads, etc.;

e) relevant modes of operation of the machine;

f) requirements on diagnostics for hardware including the characteristics of sensors, final actuators, etc.;

g) effects of mechanical tolerances, e.g. of sensors and/or their sensing counter parts;

h) coding guidelines.

8.3.2.3 Software design specification – SW level 1

The software design specification shall be derived from the software safety requirements of the SCS.

The software design specification shall be:

– structured, reviewable, testable, understandable, maintainable and operable;

– developed for each subsystem on the basis of the SCS specification and architecture;

– sufficiently detailed to allow the design and implementation of the SCS to achieve the required level of safety (SIL), and to allow verification and testing;

– traceable back to the specification of the software safety requirements of the SCS. This means that the specification is as such understandable such that another person (e.g. non-software specialist) can verify if the specification corresponds to the software safety requirements of the safety functions defined in the risk assessment;

– free of ambiguous terminology and irrelevant descriptions.

It shall be possible to relate the inputs of the software design specification in a straightforward manner to the desired outputs and vice versa. Where appropriate, easy readable semi-formal methods such as cause&effect tables, logic tables or diagrams, function-blocks or sequence diagrams shall be used in the documentation.

NOTE 1 Where appropriate depends on the number of safety functions involved in the program. Whenever the total amount of safety functions inside the program is larger than 3, it is considered appropriate.

The following shall be specified within the software design specification:

a) logic of the safety functions, including safety-related inputs and outputs and proper diagnostics on detected faults. Possible methods include, but are not limited to, cause&effect table, written description or function blocks;

NOTE 2 Faults can also be detected by hardware (e.g. signal discrepancy detected by input card).

b) test cases that include:
   – the specific input value(s) for which the test is carried out and the expected test results including pass/fail criteria;
   – fault insertion or injection(s);

NOTE 3 For simple functions, the test case(s) can be given implicitly by the specification of the safety function.

c) diagnostic functions for input devices, such as sensing elements and switches, and final control elements, such as solenoids, relays, or contactors;

d) functions that enable the machine to achieve or maintain a safe state;

e) functions related to the detection, annunciation and handling of faults;

f) functions related to the periodic testing of SCS(s) on-line and off-line;

g) functions that prevent unauthorized modification of the SCS (e.g. password);

h) interfaces to non SCS;

i) safety function response time.

NOTE 4 Guidance on software documentation is given in IEC 61508, ISO/IEC/IEEE 26512.

The software design specification shall also explain the main software aspects. Main aspects include for example:

– if appropriate, the software architecture that defines the structure decided to satisfy the software design specification;
– the global data;
– data libraries used;
– pre-existing software modules used;
– diagnostic functions (internal, external);
– programming tools including information which uniquely identifies the tool;
– integration test cases and procedures, including specification of the test environment, support software, configuration description and procedures for corrective action on failure of test.

It is recommended to use pre-designed software modules within the software design specification wherever possible, for example a software module used for muting function according to IEC 61496-1 and designed by the manufacturer of the platform.

It is recommended that in case of pre-designed safety sub-functions, for example IEC 61800-5-2, a reference to the specification provided by the manufacturer should be used.

The information in the software design specification shall be reviewed and where necessary revised, to ensure that the software safety requirements (see 8.3.2.2) are adequately specified.

8.3.3 Module design – SW level 1

8.3.3.1 General – SW level 1

Where previously developed software library modules are to be used as part of the design, their suitability in satisfying the specification of requirements of the software safety shall be demonstrated. Constraints from the previous software development environment (for example operating system and compiler dependencies) shall be evaluated.

8.3.3.2 Input information – SW level 1

For software modules, the following information shall be available in the module requirements:

a) module description;

b) module interface (inputs and outputs with data types, and, if necessary, with data ranges);

c) module libraries used;

d) specific coding rules.

8.3.3.3 Module design specification – SW level 1

The module design specification shall contain the following information:

a) description of the logic (i.e. the functionality) of each module;

b) fully defined input and output interfaces assigned for each module;

c) format and value ranges of input and output data and their relation to modules;

d) test cases which shall include normal and outside normal operation.

NOTE Although test cases usually comprise the individually testing of parameters within their specified ranges, a varying combination of these parameters can introduce unpredicted operation.

This information shall be reviewed against the input information (see 8.3.3.2).

8.3.4 Coding – SW level 1

Software shall be developed in accordance with the design specifications and coding rules. Coding rules can be either well-known industry standards or can be internal to the manufacturer. The code shall be reviewed against the design specifications and coding rules.

NOTE Coding rules are intended to restrict the freedom of programming in order to avoid the program code becoming incomprehensible and in order to reduce the likelihood of the program entering unintended states.

The output of coding shall comprise

– source code listing (e.g. ladder, function blocks, models);
– code review report.

Some typical coding rules to be applied, include, but are not limited to:

– Structure of the program is as easy and clear as possible.
– Structure of the program should be such that the logical flow starts at the top and follows in the effective sequence.
– Every part should have sufficient comments in a predefined way.
– Same names for parameters as during design should be used.
– Names should represent the function of the parameter in a clear way.
– A predefined state should exist.
– Use of set/reset for safety functions should be limited.
– Safety outputs should only be assigned once inside a program.
– Non safety parameters shall not be used to bypass safety functions.

8.3.5 Module test – SW level 1

Each module which was not previously assessed shall be tested against the test cases defined in the module design by functional and black-box, grey-box or white-box testing as appropriate.

If the module does not pass the testing, then predefined corrective action shall be taken.

The test results, and corrective actions, shall be documented.

NOTE 1 Functional testing aims to reveal failures during the specification and design phases, and to avoid failures during implementation and the integration of software and hardware.

NOTE 2 Black-box testing aims to check the dynamic behaviour under real functional conditions, and to reveal failures to meet functional specification, and to assess utility and robustness. Grey-box testing is similar to Black-box testing but additionally monitors relevant test parameter(s) inside the software module.

8.3.6 Software testing – SW level 1

8.3.6.1 General – SW level 1

The main goal of software testing is to ensure that the functionality as detailed in the software design specification is achieved.

The main output of software testing is a document e.g. a test report with test cases and test results allowing an assessment of the test coverage.

Software testing shall also include failure simulation and the associated failure reaction depending on the required safety integrity.

When pre-designed input cards or software modules which incorporate failure detection and reaction are utilised (e.g. discrepancy of input signals or feedback contact of output) then the test of those failure detection and reaction is not necessary. In that case, only the integration of these input cards or software modules in accordance with the manufacturer's specification shall be tested.

Software testing can be carried out as part of the system validation if testing is performed on the target hardware.

Functional testing as a basic measure shall be applied. Code should be tested by simulation where feasible.

It is recommended to define general guidelines or procedures for the testing of safety-related software. These guidelines or procedures should include:

– types of tests to be performed;
– specification of test equipment including tools, support software and configuration description;
– management of software versioning during testing and correcting of safety-related software;
– corrective actions on failed test;
– criteria for the completion of the test with respect to the related functions or requirements; physical location(s) of the testing, such as computer simulation, bench top or lab, factory, or on the machine.

8.3.6.2 Test planning and execution – SW level 1

Test planning based on test cases shall include:

– definition of roles and responsibilities by name;
– installation testing;
– functional testing.

8.3.7 Documentation – SW level 1

All life cycle activities shall be traceable forwards and backwards from the specification of the safety function(s) and through the completed validation plan.

The inputs and outputs of all software safety lifecycle phases shall be documented and made available to the relevant persons.

The test activities results and corrective actions taken shall be documented.

8.3.8 Configuration and modification management process – SW level 1

Any modifications or changes to software shall be subject to an impact analysis that identifies all software parts affected and the necessary re-design, re-review and re-test activities to confirm that the relevant software safety requirements are still satisfied.

Configuration management processes and modifications management processes shall be defined and documented. This shall, as a minimum, include the following items:

– articles managed by the configuration, at least: software safety requirements preliminary and detailed software design, source code modules, plans, procedures and results of the validation tests;

– identification rules which uniquely identify each software module or configuration element;

– modification processes which are comprehensive from request through to implementation.

For each article of configuration, it shall be possible to identify any changes that can have occurred and the versions of any associated elements.

NOTE 1 The purpose is to be able to trace the historical development of each article: what modifications have been made, why, and when.

Software configuration management shall allow a precise and unique software version identification to be obtained. Configuration management should associate all the articles (and their version) needed to demonstrate the functional safety.

All articles in the software configuration shall be covered by the configuration management procedure before being tested or being requested by the analyst for final software version evaluation.

NOTE 2 The objective here is to ensure the evaluation procedure is performed on software with all elements in a precise state. Any subsequent change can necessitate revision of the software so that it can be identifiable by the analyst.

Procedures for the archiving of software and its associated data shall be established (methods for storing backups and archives).

NOTE 3 These backups and archives can be used to maintain and modify software during its functional lifetime.

8.4 Software level 2

8.4.1 Software safety lifecycle – SW level 2

8.4.1.1 Maximum achievable SIL – SW level 2

The maximum achievable SIL for SW level 2 is SIL 2.

8.4.1.2 Software safety lifecycle model – SW level 2

A software safety lifecycle model which is resolved into distinct phases shall be used (e.g. V-model), including management and documentation activities to achieve the required level of safety.

Any software lifecycle model may be used provided all the objectives and requirements of this Subclause 8.4 are met. Safety-related software shall be validated as described in ~~9.5.3~~ 9.5.4.

Software level 2 is of increased complexity in comparison with SW level 1 due to the use of fully variable programming languages. Therefore, the more detailed V-model in Figure 14 is applicable.

NOTE 1 The V-model is a static model used to structure the software design into small parts. It does not introduce any sequence of creation of specifications or implementation. The left side represents requirements; i.e. things to achieve. The right side details testing of the software.

NOTE 2 On the left side of the V-model, the output of each phase is reviewed. Review means to check the output of a phase in the V-model against the requirements of the input of the same phase. The arrow 'Review' represents the first step of the software verification. Further information on the level of independence of review and testing/verification is available in Annex J.

NOTE 3 Project management techniques and processes can be chosen to be appropriate for the size and scope of the project.

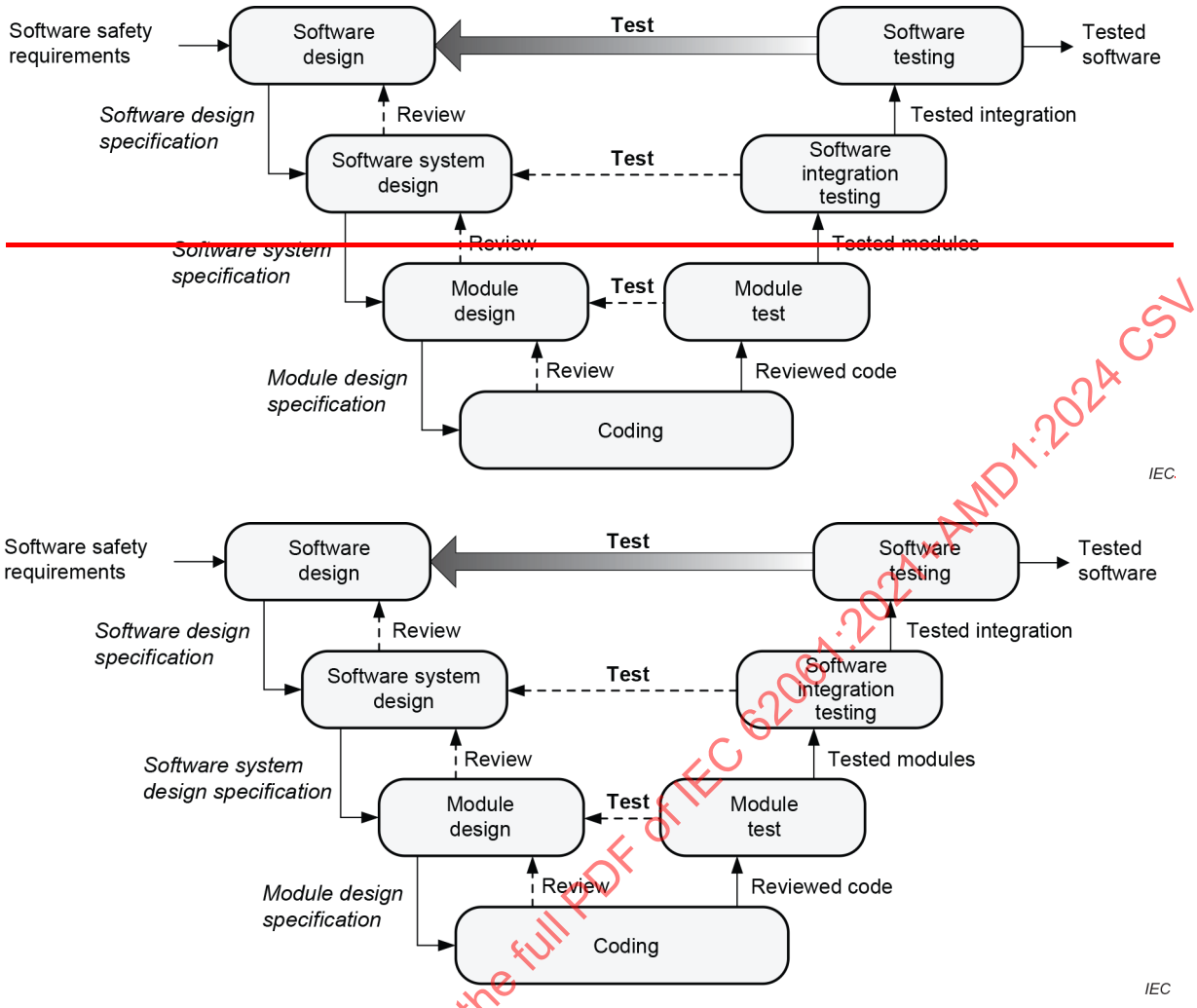


Figure 14 – V-model of software safety lifecycle for SW level 2

NOTE 4 In the V-models, the arrow 'Test' represents the results of test cases according to the specification and, in addition, the need for more precise test case requirements and specifications.

8.4.1.3 Tools usage – SW level 2

A suitable set of tools shall be selected (e.g. configuration management, simulation, and test equipment with test generator). Preferably, the recommended tools from the manufacturer should be applied. The availability of suitable tools for service, updating the machine, and parameterization over the lifetime of the safety-related control system shall be considered. Either the tools provided by the manufacturer of the equipment are used or the suitability of the tools shall be explained and documented.

Suitability shall be proven as follows:

– an analysis carried out to identify possible effects of a failure caused by these tools in the tool chain; and

– appropriate fault avoiding and fault controlling measures be selected, applied, and their effectiveness be verified via rigorous testing and the results documented.

NOTE 1 The appropriateness of fault avoiding and fault controlling measures depends on the severity of the consequence of a failure. The basis of this evaluation is an analysis. To carry out this analysis, it is necessary to have knowledge regarding the application of the support tool and of the machine.

NOTE 2 The effect of failures can vary between different support tools. Therefore IEC 61508-4 differentiates between three categories for off-line support tools used within the software development lifecycle. This can be part of the analysis.

NOTE 3 See IEC 61508-4 for definition of support tools and examples.

NOTE 4 This document does not specify any measures for avoiding or controlling faults of off-line support tools. For examples, see IEC 61508-3:2010, 7.4.4.

8.4.2 Software design – SW level 2

8.4.2.1 General – SW level 2

The software design specification shall be developed on basis of the software safety requirements and managed throughout the lifecycle of the SCS.

8.4.2.2 Software safety requirements – SW level 2

To support the software design process, the following information shall be considered:

a) specification of the safety function(s) (see 5.2);

b) configuration or architecture of the SCS (e.g. hardware architecture, wiring diagram, safety-related inputs and outputs);

c) response time requirements;

d) operator interfaces and controls, such as: switches, joysticks, mode selector, dials, touch sensitive control devices, keypads, etc.;

e) relevant modes of operation of the machine;

f) requirements on diagnostics for hardware including the characteristics of sensors, final actuators, etc.;

g) effects of mechanical tolerances, e.g. of sensors and/or their sensing counter parts;

h) coding guidelines.

When applying SW level 2, the tables of IEC 61508-3:2010, Annex A and Annex B shall be taken into consideration when it is appropriate to use alternative techniques and measures of an equivalent effectiveness. IEC 61508-7 provides additional information.

The design and choice of the language chosen to satisfy the required SIL of the SCS shall be appropriate for the application.

The design shall include self-monitoring of control flow and data flow appropriate to the SIL of the SCS. On failure detection, appropriate actions shall be performed to achieve or maintain a safe state.

8.4.2.3 Software design specification – SW level 2

The software design specification shall be derived from the software safety requirements of the SCS.

The software design specification shall be:

– structured, reviewable, testable, understandable, maintainable and operable;

– developed for each subsystem on the basis of the SCS specification and architecture;

– sufficiently detailed to allow the design and implementation of the SCS to achieve the required level of safety (SIL), and to allow verification and testing.

– traceable back to the specification of the software safety requirements of the SCS. This means that the specification is as such understandable such that another person (e.g. non-software specialist) can verify if the specification corresponds to the software safety requirements of the safety functions defined in the risk assessment.

– free of ambiguous terminology and irrelevant descriptions.

It shall be possible to relate the inputs of the software design specification in a straightforward manner to the desired outputs and vice versa. Where appropriate, easily readable semi-formal methods such as cause&effect tables, logic tables or diagrams, function-blocks or sequence diagrams shall be used in the documentation.

NOTE 1 Where appropriate depends on the number of safety functions involved in the program. Whenever the total amount of safety functions inside the program is larger than 3, it is considered appropriate.

The following shall be specified within the software design specification:

a) logic of the safety functions, including safety-related inputs and outputs and proper diagnostics on detected faults. Possible methods include, but are not limited to, cause&effect table, written description or function blocks;

NOTE 2 Faults can also be detected by hardware (e.g. signal discrepancy detected by input card).

b) test cases that include:
   - the specific input value(s) for which the test is carried out and the expected test results including pass/fail criteria;
   - fault insertion or injection(s).

NOTE 3 For simple functions, the test case(s) can be given implicitly by the specification of the safety function.

c) diagnostic functions for input devices, such as sensing elements and switches, and final control elements, such as solenoids, relays, or contactors;
d) functions that enable the machine to achieve or maintain a safe state;
e) functions related to the detection, annunciation and handling of faults;
f) functions related to the periodic testing of SCS(s) on-line and off-line;
g) functions that prevent unauthorized modification of the SCS (e.g. password);
h) interfaces to non SCS;
i) safety function response time.

NOTE 4 Guidance on software documentation is given in IEC 61508, ISO/IEC/IEEE 26512.

It is recommended to use pre-designed software modules within the software design specification wherever possible.

It is recommended that in case of pre-designed safety sub-functions, for example IEC 61800-5-2, a reference to the specification provided by the manufacturer should be used.

The information in the software design specification shall be reviewed and where necessary revised, to ensure that the requirements of the software safety requirements (see 8.4.2.2) are adequately specified.

8.4.3 Software system design – SW level 2

8.4.3.1 General – SW level 2

Software system design starts with architecture definition. Software architecture shall be established that fulfils the software design specification. The software architecture defines the major elements and subsystems of the software, how they are interconnected and how the required attributes will be achieved. It also defines the overall behavior of the software, and how software elements interface and interact. Examples of major software elements include operating systems, databases, input/output subsystems, communication subsystems, application programs, programming and diagnostic tools, etc.

The software system design shall follow a modular approach with a limited software module size, a fully defined interface and one entry/one exit point in subroutines and functions. Each module shall have a single, clearly understood function or purpose. The maximum module size shall be limited to one complete safety function.

The following programming techniques shall be used to avoid systematic failures:

– range checking and plausibility checking of variables and configuration parameters;
– temporal or logical program sequence monitoring to detect a defective program sequence: A defective program sequence exists if the individual elements of a program (e.g. software modules, subprograms or commands) are processed in the wrong sequence or period of time or if the clock of the processor is faulty (see IEC 61508-7:2010, Clause A.9);
– limiting the number or extent of global variables.

NOTE For Software level 2, see Annex G of IEC 61508-7:2010 for guidance on object oriented architecture and design.

8.4.3.2 Software system design specification – SW level 2

A software system design specification shall be provided as an output of the software system design. This shall explain the main software aspects such as indicated in the following list, for example:

– the software architecture that defines the structure decided to satisfy the software design specification;
– the global data;
– data libraries used;
– pre-existing software modules used;
– diagnostic functions (internal, external);
– programming tools including information which uniquely identifies the tool;
– integration test cases and procedures, including specification of the test environment, support software, configuration description and procedures for corrective action on failure of test.

The information contained in the software system specification shall be reviewed against the software design specification.

8.4.4 Module design – SW level 2

8.4.4.1 General – SW level 2

Where previously developed software library modules are to be used as part of the design, their suitability in satisfying the specification of requirements of the software safety shall be demonstrated. Constraints from the previous software development environment (for example operating system and compiler dependencies) shall be evaluated.

8.4.4.2 Input information – SW level 2

For software modules, the following information shall be available in the software system design specification:

a) module description;
b) module interface (inputs and outputs with data types and, if necessary, with data ranges);
c) module libraries used;
d) special coding rules.

8.4.4.3 Module design specification – SW level 2

The module design specification shall contain the following information:

a) description of the logic (i.e. the functionality) of each module;
b) fully defined input and output interfaces assigned for each module;

c) format and value ranges of input and output data and their relation to modules;

d) test cases which shall include normal and outside normal operation;

NOTE Although test cases usually comprise the individual testing of parameters within their specified ranges, a varying combination of these parameters can introduce unpredicted operation.

e) documentation of the interrupts.

This information shall be reviewed against the input information (see 8.4.4.2).

8.4.5 Coding – SW level 2

Software shall be developed in accordance with the design specifications and coding rules. Coding rules can be either well-known industry standards or can be internal to the manufacturer. The code shall be reviewed against the design specifications and coding rules.

NOTE 1 Coding rules are intended to restrict the freedom of programming in order to avoid the program code becoming incomprehensible and in order to reduce the likelihood of the program entering unintended states.

NOTE 2 Coding rules usually define a subset of a programming language or use of a strongly typed programming language (see IEC 61508-7:2010, C.4.1).

The output of coding shall comprise

– source code listing (e.g. ladder, function blocks, models);
– code review report.

Some typical coding rules to be applied, include, but are not limited to the following:

– Structure of the program is as easy and clear as possible.
– Structure of the program should be such that the logical flow starts at the top and follows in the effective sequence.
– Every part should have sufficient comments in a predefined way.
– Same names for parameters as during design should be used.
– Names should represent the function of the parameter in a clear way.
– A predefined state should exist.
– Use of set/reset for safety functions should be limited.
– Safety outputs should only be assigned once inside a program.
– Non safety parameters shall not be used to bypass safety functions.

8.4.6 Module test – SW level 2

Each module which was not previously assessed shall be tested against the test cases defined in the module design by functional and black-box, grey-box or white-box testing as appropriate.

If the module does not pass the testing, then predefined corrective action shall be taken.

The test results and corrective actions shall be documented.

NOTE 1 Functional testing aims to reveal failures during the specification and design phases, and to avoid failures during implementation and the integration of software and hardware.

NOTE 2 Black-box testing aims to check the dynamic behaviour under real functional conditions, and to reveal failures to meet functional specification, and to assess utility and robustness. Grey-box testing is similar to Black-box testing but additionally monitors relevant test parameter(s) inside the software module.

Module testing shall use as a minimum dynamic analysis and testing.

8.4.7 Software integration testing SW level 2

The software shall be tested against the integration test cases. The results of software integration testing shall be documented.

NOTE The objective of these tests is to show that all software modules and software elements/subsystems interact correctly to perform their intended function and do not perform unintended functions. This does not imply testing of all input combinations, nor of all output combinations. Testing all equivalence classes or structure based testing can be sufficient. Boundary value analysis or control flow analysis can reduce the test cases to an acceptable number. Analysable programs make the requirements easier to fulfil.

8.4.8 Software testing SW level 2

8.4.8.1 General – SW level 2

The main goal of software testing is to ensure that the functionality as detailed in the software design specification is achieved.

NOTE This can imply testing of all input combinations, and/or all output combinations.

The main output of software testing is a document, e.g. a test report with test cases and test results allowing an assessment of the test coverage.

Software testing shall also include failure simulation and the associated failure reaction depending on the required safety integrity.

When pre-designed input cards or software modules which incorporate failure detection and reaction are utilised (e.g. discrepancy of input signals or feedback contact of output) then the test of those failure detection and reaction is not necessary. In that case, only the integration of these input cards or software modules in accordance with the manufacturer's specification shall be tested.

Software testing can be carried out as part of the system validation if testing is performed on the target hardware.

Functional testing as a basic measure shall be applied. Code should be tested by simulation where feasible.

It is recommended to define general guidelines or procedures for the testing of safety-related software. These guidelines or procedures should include:

– types of tests to be performed;
– specification of test equipment including tools, support software and configuration description;
– management of software versioning during testing and correcting of safety-related software;
– corrective actions on failed test;
– criteria for the completion of the test with respect to the related functions or requirements; physical location(s) of the testing, such as computer simulation, bench top or lab, factory, or on the machine.

8.4.8.2 Test planning and execution – SW level 2

Test planning based on test cases shall include:

– definition of roles and responsibilities by name;
– installation testing;
– functional testing.

Testing of software includes two types of activities:

– Static analysis: Analysis of software documentation, e.g. by review, inspection, walk-through, control flow analysis, or dataflow analysis.

– Dynamic testing: Execution of the software in a controlled and systematic way, so as to demonstrate the presence of the required behaviour and the absence of unwanted behaviour. This includes, in particular, functional testing, black-box or grey-box-testing.

In the early phases of the software lifecycle, verification is static. Dynamic testing becomes possible when code is produced. For verifying the output of software lifecycle activities, both activities are required in combination. For further description of static analysis and dynamic testing, see IEC 61508-3.

The following is required for verification and testing of safety-related software:

– static analysis shall be done and documented in any case;

– dynamic testing shall be done and documented;

– where software is required for a safety function of up to SIL 1 and is not subject to dynamic testing, this shall be justified with respect to the structural simplicity of the software;

– for dynamic testing, every subprogram (subroutine or function) shall have been called at least once (entry points) during testing;

– for software which is required for a safety function of SIL 2, all statements in the code shall be executed at least once during dynamic testing;

– where software is used in diagnostic functions for controlling random hardware failures, dynamic testing shall address the correct implementation of the diagnostics, e.g. by fault insertion testing;

– dynamic testing shall include a final test on the target hardware.

8.4.9 Documentation – SW level 2

All life cycle activities shall be traceable forwards and backwards from the specification of the safety function(s) and through the completed validation plan.

The inputs and outputs of all software safety lifecycle phases shall be documented and made available to the relevant persons.

The test activities results and corrective actions taken shall be documented.

8.4.10 Configuration and modification management process – SW level 2

Any modifications or changes to software shall be subject to an impact analysis that identifies all software parts affected and the necessary re-design, re-review and re-test activities to confirm that the relevant software safety requirements are still satisfied.

Configuration management processes and modifications management processes shall be defined and documented. This shall, as a minimum, include the following items:

– articles managed by the configuration, at least: software safety requirements, preliminary and detailed software design, source code modules, plans, procedures and results of the validation tests;

– identification rules which uniquely identify each software module or configuration element;

– modification processes which are comprehensive from request through to implementation.

For each article of configuration, it shall be possible to identify any changes that can have occurred and the versions of any associated elements.

NOTE 1 The purpose is to be able to trace the historical development of each article: what modifications have been made, why, and when.

Software configuration management shall allow a precise and unique software version identification to be obtained. Configuration management shall associate all the articles (and their version) needed to demonstrate the functional safety.

All articles in the software configuration shall be covered by the configuration management procedure before being tested or being requested by the analyst for final software version evaluation.

NOTE 2 The objective here is to ensure that the evaluation procedure be performed on software with all elements in a precise state. Any subsequent change can necessitate revision of the software so that it can be identifiable by the analyst.

Procedures for the archiving of software and its associated data shall be established (methods for storing backups and archives).

NOTE 3 These backups and archives can be used to maintain and modify software during its functional lifetime.

9 Validation

9.1 Validation principles

In this document, the purpose of the validation is to confirm that the SCS complies with the safety requirements specification given in Clause 5 and the information for use in 10.3.

NOTE 1 In this document, the validation is limited to the designed SCS or a part of it supporting the safety functions required from the risk reduction strategy at the machine level given in ISO 12100. The SCS validation result is intended to be part of the overall validation of the machine.

NOTE 2 In some cases, the safety validation can only be completed after final installation (for example, when the application software development is not finalized).

The validation activities consist of collecting and checking the availability of the evidence demonstrating the completeness of each design activity identified in the safety plan.

The validation to be applied to the SCS includes inspection (e.g. by analysis) and testing of the SCS to ensure that it achieves the requirements stated in the safety requirements specification (according to Clause 5).

The validation shall demonstrate that the SCS meets the requirements and, in particular, the following:

a) the specified functional requirements of the safety functions provided by that part (see 5.2), as set out in the design rationale;

b) the requirements of the specified SIL.

Validation shall be carried out by persons who are independent from the design of the SCS.

NOTE 3 “Independent person” does not necessarily mean that a third-party test is required.

The analysis should be started as early as possible in, and in parallel with, the design process.

NOTE 4 Problems can then be corrected early while they are still relatively easy to correct, i.e. during steps “design and technical realization of the safety function” and “evaluate the SIL”. It can be necessary for some parts of the analysis to be delayed until the design is well developed.

Figure 15 gives an overview of the validation process: validation consists of applying analysis (see 9.2) and executing functional tests (see 9.3) under foreseeable conditions in accordance with the validation plan. The balance between the analysis and testing shall be justified. For architectures with diagnostic function, the validation of the safety function shall also include testing under fault conditions to show that the fault reaction will be initiated by the implemented diagnostic function.

Where appropriate due to the system's size, complexity or the effects of integrating it with the control system (of the machinery), special arrangements should be made for

– validation of the subsystem separately before integration, including simulation of the appropriate input and output signals, and
– validation of the effects of integrating safety-related parts into the remainder of the control system within the context of its use in the machine.

"Modification of the design" in Figure 15 refers to the design process. If the validation cannot be successfully completed, changes in the design are necessary. The validation of the SCS should then be repeated as appropriate. This process should be iterated until the SCS for each safety function is successfully validated.

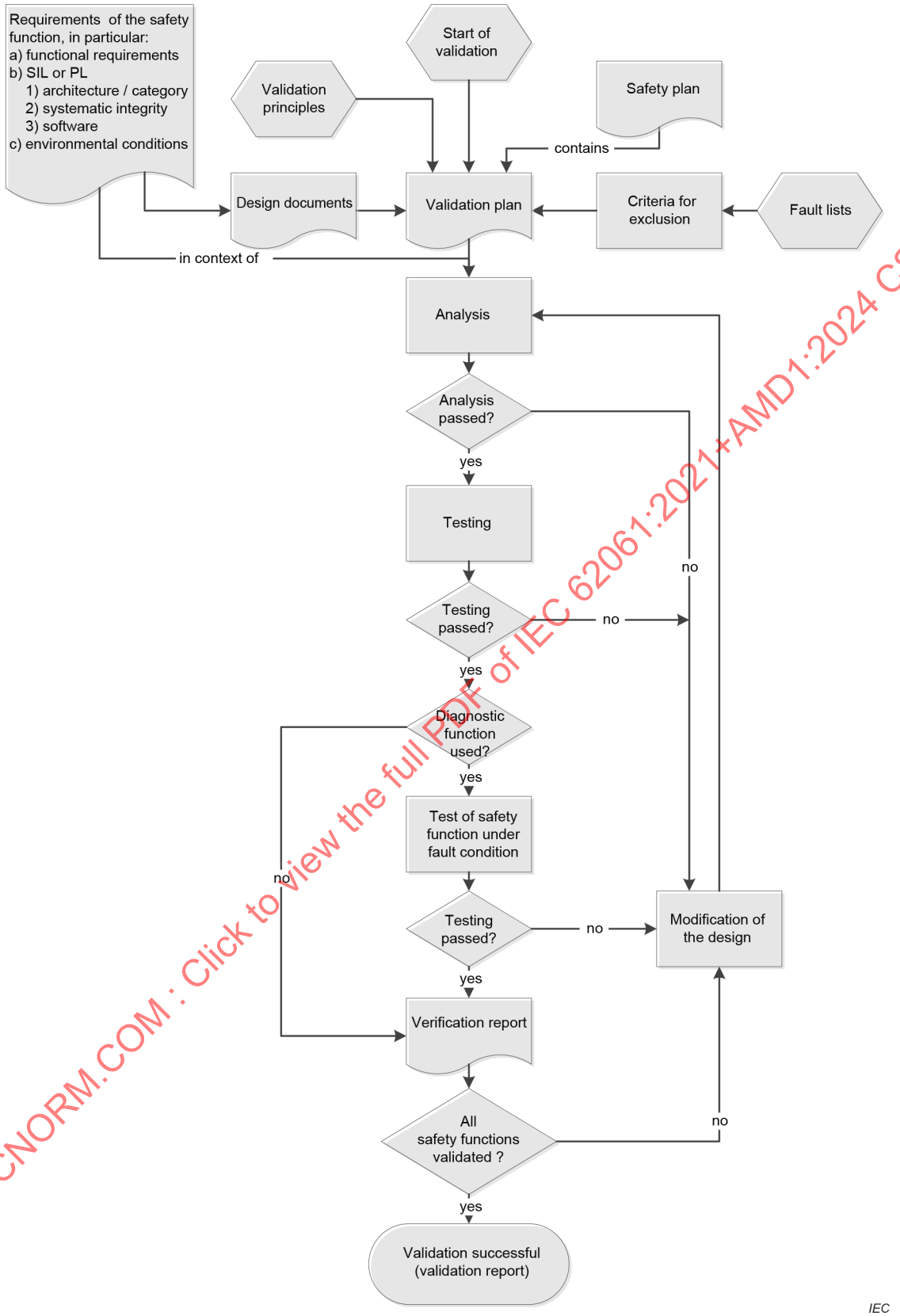


Figure 15 – Overview of the validation process

9.1.1 Validation plan

The validation plan shall identify and describe the requirements for carrying out the validation process. The validation plan shall also identify the means to be employed to validate the specified safety functions. It shall set out, where appropriate:

a) the identity of the specification documents,
b) the operational and environmental conditions during testing,
c) the analyses and tests to be applied,
d) the reference to test standards to be applied,
e) the persons or parties responsible for each step in the validation process, and
f) the required equipment.

Subsystems which have previously been validated to the same specification need only reference to that previous validation

NOTE Information on the level of independence of validation is available in Annex J.

9.1.2 Use of generic fault lists

Validation involves consideration of the behaviour of the SCS for all faults to be considered. A basis for fault consideration is given in the tables of fault lists of ISO 13849-2:2012, Annexes A to D, which are based on experience and which contain:

– the components/elements to be included, e.g. conductors/cables,
– the faults to be taken into account, e.g. short circuits between conductors,
– the permitted fault exclusions, taking into account environmental, operating and application aspects, and
– a remarks section giving the reasons for the fault exclusions.

Only permanent faults are taken into account in the fault lists.

9.1.3 Specific fault lists

If necessary, a specific product-related fault list shall be generated as a reference document for the validation of the subsystem(s) and/or subsystem element(s).

NOTE The list can be based on the appropriate generic list(s) found in the annexes A to D in ISO 13849-2:2012.

Where the specific product-related fault list is based on the generic list(s), it shall state

a) the faults taken from the generic list(s) to be included,
b) any other relevant faults to be included but not given in the generic list (e.g. common-cause failures),
c) the faults taken from the generic list(s) which may be excluded on the basis that the criteria given in the generic list(s) are satisfied (see 7.3.3),
d) and exceptionally any other faults for which the generic list(s) do not permit an exclusion, but for which justification and rationale for an exclusion is presented (see 7.3.3).

Where this list is not based on the generic list(s), the designer shall give the rationale for fault exclusions.

9.1.4 Information for validation

The information required for validation will vary with the technology used, the architectural constraints and SIL to be demonstrated, the design rationale of the system, and the contribution of the SCS to the reduction of the risk. Documents containing sufficient information from the following list shall be included as appropriate in the validation to demonstrate that the safety-related parts perform the specified safety functions to the required SIL and architectural constraints:

a) specification of the required characteristics of each safety function, especially the required SIL and architectural constraints;

b) drawings and specifications, e.g. for mechanical, hydraulic and pneumatic parts, printed circuit boards, assembled boards, internal wiring, enclosure, materials, mounting;

c) block diagram(s) with a functional description of the blocks;

d) circuit diagram(s), including interfaces/connections;

e) functional description of the circuit diagram(s);

f) time sequence diagram(s) for switching components, signals relevant for safety;

g) description of the relevant characteristics of components previously validated;

h) for safety-related parts other than those listed in g), component lists with item designations, rated values, tolerances, relevant operating stresses, type designation, failure-rate data and component manufacturer, and any other data relevant to safety;

i) analysis of all relevant faults according to 9.1.2 and 9.1.3, such as those listed in the tables of ISO 13849-2:2012, Annexes A to D, including the justification of any excluded faults;

j) an analysis of the influence of processed materials;

k) information for use, e.g. installation and operation manual/instruction handbook.

Where software is relevant to the safety function(s), the software documentation shall include

– a specification which is clear and unambiguous and which states the safety integrity the software is required to achieve,

– evidence that the software is designed to achieve the required SIL (see 9.5.4), and

– details of the verification (in particular test reports) carried out to prove that the required SIL is achieved.

Information is required on how the SIL and PFH is determined. The documentation of the quantifiable aspects shall include

– the basic subsystem architecture according to 7.5.2,

– the determination reliability parameters (e.g. $MTTF_D$ or λ_D of subsystem elements and CCF), and

– the determination of the architectural constraints.

Information is required for documentation on systematic aspects of the SCS. Information is required to describe how the combination of several subsystems achieves a SIL in accordance with the required SIL.

9.1.5 Validation record

Validation by analysis and testing shall be recorded, see also Clause 10. Appropriate documentation shall state:

- the version of the validation plan being used, and the version of the safety function tested;
- the safety function under test (or analysis), along with the specific reference to the requirement specified during the validation planning;
- referenced standards;

- tools and equipment used, along with calibration data;
- the results of each test;
- discrepancies between expected and actual results.

9.2 Analysis as part of validation

9.2.1 General

Validation of the SCS shall be carried out by analysis. Inputs to the analysis include the following:

– the safety function(s), their characteristics and the safety integrity specified according to Clause 5;

– the system structure (e.g. basic subsystem architectures) according to 7.5.2;

– the quantifiable aspects (e.g. $MTTF_D$ or λ_D, DC and CCF) according to 6.4.2;

– the non-quantifiable, qualitative aspects which affect system behaviour (if applicable, software aspects);

– deterministic arguments.

NOTE 1 A deterministic argument is an argument based on qualitative aspects (e.g. quality of manufacture, experience of use). This consideration depends on the application, which, together with other factors, can affect the deterministic arguments.

NOTE 2 Deterministic arguments differ from other evidence in that they show that the required properties of the system follow logically from a model of the system. Such arguments can be constructed on the basis of simple, well-understood concepts.

9.2.2 Analysis techniques

The selection of an analysis technique depends upon the particular object. Two basic techniques exist, as follows.

a) Top-down (deductive) techniques are suitable for determining the initiating events that can lead to identified top events, and calculating the probability of top events from the probability of the initiating events. They can also be used to investigate the consequences of identified multiple faults.

EXAMPLE Fault tree analysis (FTA, see IEC 61025), event tree analysis (ETA, see IEC 62502).

b) Bottom-up (inductive) techniques are suitable for investigating the consequence of identified single faults.

EXAMPLE Failure modes and effects analysis (FMEA, see IEC 60812) and failure modes, effects and criticality analysis (FMECA).

9.2.3 Verification of safety requirements specification (SRS)

The requirements specification for the safety function shall be verified to ensure consistency and completeness and correctness for its intended use.

Verification may be performed by reviews and inspections of the SCS safety requirements and design specification(s), in particular to prove that all aspects of

– the intended application requirements and safety needs, and

– the operational and environmental conditions and possible human errors (e.g. misuse) have been considered.

9.3 Testing as part of validation

9.3.1 General

Testing shall be carried out to complete the validation. Validation tests shall be planned and implemented in a logical manner. In particular:

a) a test plan shall be produced before testing begins that shall include
   1) the test specifications;
   2) the required outcome of the tests for compliance, and
   3) the chronology of the tests;
b) test records shall be produced that include
   4) the name of the person carrying out the test;
   5) the environmental conditions;
   6) the test procedures and equipment used;
   7) the date of the test, and
   8) the results of the test;
c) the test records shall be compared with the test plan to ensure that the specified functional and performance targets are achieved.

The test sample shall be operated as near as possible to its final operating configuration.

This testing can be applied manually or automatically, e.g. by computer.

Where applied, validation of the safety functions by testing shall be carried out by applying input signals, in various combinations, to the SCS. The resultant response at the outputs shall be compared to the appropriate specified outputs.

It is recommended that the combination of these input signals be applied systematically to the control system and the machine. An example of this logic is power-on, start-up, operation, directional changes, restart-up. Where necessary, an expanded range of input data shall be applied to take into account anomalous or unusual situations, in order to see how the SCS responds. Such combinations of input data shall take into account foreseeable incorrect operation(s).

The objectives of the test will determine the environmental condition for that test, which can be one or another of the following:

– the environmental conditions of intended use;
– the conditions at a particular rating;
– a given range of conditions if drift is expected.

9.3.2 Measurement accuracy

The accuracy of measurements during the validation by testing shall be appropriate for the test carried out. In general, these measurement accuracies shall be within 5 K for temperature measurements and 5 % for the following:

a) time measurements;
b) pressure measurements;
c) force measurements;
d) electrical measurements;
e) relative humidity measurements;

f) linear measurements.

Deviations from these measurement accuracies shall be justified.

9.3.3 More stringent requirements

If, according to its accompanying documentation, the requirements for the SCS exceed those within this document, the more stringent requirements shall apply.

NOTE More stringent requirements can apply if the control system has to withstand particularly adverse service conditions, e.g. rough handling, humidity effects, hydrolysation, ambient temperature variations, effects of chemical agents, corrosion, high strength of electromagnetic fields – for example, due to close proximity of transmitters.

9.3.4 Test samples

Test samples shall not be modified during the course of the tests.

Certain tests can permanently change the performance of some components. Where a permanent change in a component causes the safety-related part to be incapable of meeting the requirements of further tests, a new sample or samples shall be used for subsequent tests.

Where a particular test is destructive and equivalent results can be obtained by testing part of SCS in isolation, a sample of that SCS may be used instead of the whole SCS for the purpose of obtaining the results of the test. This approach shall only be applied where it has been shown by analysis that testing of a part of SCS is sufficient to demonstrate the safety integrity of the whole SCS that performs the safety function.

9.4 Validation of the safety function

9.4.1 General

The validation of safety functions shall demonstrate that the SCS provides the safety function(s) in accordance with their specified characteristics.

NOTE 1 A loss of the safety function in the absence of a hardware fault is due to a systematic fault, which can be caused by errors made during the design and integration stages (a misinterpretation of the safety function characteristics, an error in the logic design, an error in hardware assembly, an error in typing the code of software, etc.). Some of these systematic faults will be revealed during the design process, while others will be revealed during the validation process or will remain unnoticed. In addition, it is also possible for an error to be made (e.g. failure to check a characteristic) during the validation process.

Validation of the specified characteristics of the safety functions shall be achieved by the application of appropriate measures from the following list:

– functional analysis of schematics, reviews of the software (see 9.5.3);

NOTE 2 Where a machine has complex or a large number of safety functions, an analysis can reduce the number of functional tests required.

– simulation;
– check of the hardware components installed in the machine and details of the associated software to confirm their correspondence with the documentation (e.g. manufacture, type, version);
– functional testing of the safety functions in all required operating modes as defined in the SRS of the machine, to establish whether they meet the specified characteristics (see Clause 5). The functional tests shall ensure that all safety-related outputs are realized over their complete ranges and respond to safety-related input signals in accordance with the specification. The test cases are normally derived from the specifications but could also include some cases derived from analysis of the schematics or software;
– extended functional testing to check foreseeable abnormal signals or combinations of signals from any input source, including power interruption and restoration, and incorrect operations;

– check usability of the operator–SCS interface.

NOTE 3 Consider for example an HMI for software based parameterization of the safety function. In general, more information is available in IEC 60204-1 or IEC 61310 (all parts).

NOTE 4 Other measures against systematic failures mentioned in 9.5.2 (e.g. diversity, failure detection by automatic tests) can also contribute in the detection of functional faults.

9.4.2 Analysis and testing

Analysis and testing will require failure analysis using circuit diagrams and, where the failure analysis does not reach a clear result:

– fault injection tests on the actual circuit and fault initiation on actual components, particularly in parts of the system where there is doubt regarding the results obtained from failure analysis (see 9.2); or

– a simulation of control system behaviour in the event of a fault, e.g. by means of hardware and/or software models.

Fault injection or fault simulation tests can be performed at different levels, e.g. subsystem element or subsystem level, considering the specific application and test setup.

When validating by testing, the tests shall include, as appropriate,

– fault injection tests into a production sample,

– fault injection tests into a hardware model,

– software simulation of faults, and

– subsystem failure, e.g. power supplies.

The precise instant at which a fault is injected into a system can be critical. The worst-case effect of a fault injection shall be determined by analysis and by injecting the fault at this appropriate critical time.

9.5 Validation of the safety integrity of the SCS

9.5.1 General

The following steps shall be performed:

– verification for correct evaluation of SIL of the SCS based on subsystems, architecture and reliability parameters (e.g. DC and $MTTF_D$ or λ_D);

– verification that the SIL achieved by the SCS satisfies the required SIL in the safety requirements specification for the machinery: SIL ≥ required SIL.

9.5.2 Validation of subsystem(s)

The safety integrity of each subsystem of the SCS is characterized by its SIL and shall be validated by confirming (verification) the following:

– the used architecture (see 7.5.2), and

– the PFH (see 7.6), and

– the systematic integrity (see 7.3.2, Software, CCF).

In this context, the validation of $MTTF_D$ or λ_D, DC and CCF is typically performed by analysis and visual inspection. The $MTTF_D$ or λ_D values for components (including B_{10} or B_{10D}, T_{10D} and duty cycle values) shall be checked for plausibility. For example, the value given on the manufacturer's datasheet is to be compared with Annex A.

NOTE 1 A fault exclusion implies infinite $MTTF_D$; therefore, the component will not contribute to the calculation of channel $MTTF_D$.

The DC values for components (subsystem elements) and/or logic blocks shall be checked for plausibility (e.g. against measures in Annex D). The correct implementation (hardware and software) of checks and diagnostics, including appropriate fault reaction, shall be validated by testing under typical environmental conditions in use.

The correct implementation of sufficient measures against common-cause failures shall be validated (e.g. against Annex E). Typical validation measures are static hardware analysis and functional testing under environmental conditions.

NOTE 2 Generally, for the specification of the $MTTF_D$ or λ_D values of electronic components, an ambient temperature of +40 °C is taken as a basis. During validation, it is important to ensure that, for $MTTF_D$ or λ_D values, the environmental and functional conditions (in particular temperature) taken as basis are met. Where a device, or component, is operated significantly above (e.g. more than 15 °C) the specified temperature of +40 °C, it will be necessary to use $MTTF_D$ or λ_D values for the increased ambient temperature.

9.5.3 Validation of measures against systematic failures

The validation of measures against systematic failures can typically be provided by:

a) inspections of design documents which confirm the application of
   - basic and well-tried safety principles (see ISO 13849-2:2012, Annexes A to D);
   - further measures for avoidance of systematic failures, and
   - further measures for the control of systematic failures such as hardware diversity, modification protection or failure assertion programming;

b) failure analysis (e.g. FMEA);

c) fault injection tests/fault initiation;

d) inspection and testing of data communication, e.g. parameterization, installation;

e) checking that a quality management system avoids the causes of systematic failures in the manufacturing process.

9.5.4 Validation of safety-related software

The validation of software shall include:

– the specified functional behaviour and performance criteria (e.g. timing performance) of the software when executed on the target hardware,

– verification that the software measures are sufficient for the specified required SIL of the safety function, and

– measures and activities taken during software development to avoid systematic software faults.

As a first step, check that there is documentation for the specification and design of the safety-related software. This documentation shall be reviewed for completeness and absence of erroneous interpretations, omissions or inconsistencies.

NOTE In the case of small programs, an analysis of the program by means of reviews or walk-through of control flow, procedures, etc. using the software documentation (control flow chart, source code of modules or blocks, I/O and variable allocation lists, cross-reference lists) can be sufficient.

In general, software can be considered a “black box” or “grey box” (see Clause 8), and validated by the black- or grey-box test, respectively.

Depending on the required SIL, the tests should include, as appropriate,

– black-box testing of functional behaviour and performance (e.g. timing performance),

– additional extended test cases based upon limit value analyses, recommended for SIL 2 or SIL 3,
– I/O tests to ensure that the safety-related input and output signals are used properly, and
– test cases which simulate faults determined analytically beforehand, together with the expected response, in order to evaluate the adequacy of the software-based measures for control of failures.

Individual software functions which have already been validated do not need to be validated again. Where a number of such safety function blocks are combined for a specific project, however, the resulting total safety function shall be validated.

Software documentation shall be checked to confirm that sufficient measures and activities have been implemented against systematic software faults in accordance with the simplified V-model (see Figure 12).

The measures for software implementation and configuration and modification management according to Clause 8, which depend on the SIL to be attained, shall be examined with regard to their proper implementation.

Should the safety-related software be subsequently modified, it shall be revalidated on an appropriate scale.

9.5.5 Validation of combination of subsystems

Where the safety function is implemented by two or more subsystems, validation of the combination – by analysis and, if necessary, by testing – shall be undertaken to establish that the combination achieves the safety integrity specified in the design. Existing recorded validation results of subsystems can be taken into account. The following validation steps shall be performed:

– inspection of design documents describing the overall safety function(s);
– a check that the overall SIL of the subsystem combination has been correctly evaluated, based on the SIL of each individual subsystem (according to 6.4.2);
– consideration of the characteristics of the interfaces, e.g. voltage, current, pressure, data format of information, signal level;
– failure analysis relating to combination/integration, e.g. by FMEA;
– for redundant systems, fault injection tests relating to combination/integration.

10 Documentation

10.1 General

The manufacturer of an SCS and the manufacturer of subsystems shall prepare the relevant technical documentation in 10.2 and information for use in 10.3.

The documentation shall demonstrate the procedure that has been followed and the results that have been received.

The documentation shall be subject to version control.

10.2 Technical documentation

The documentation shall contain information relevant to the safety-related part:

– safety function(s) provided by the SCS according to Clause 5 or the safety sub-function provided by the SCS subsystem;

NOTE 1 Only safety functions which are required by the specific application need to be considered.

– if the design includes the subsystem design (see Clause 7), then the technical documentation shall:
  - cover the test or analysis of fault behaviour leading to a loss of the safety function or
  - refer to a qualified example (e.g. an application note);
– the characteristics of each safety function according to 5.2;
– proof test procedures when proof testing is defined for the SCS;
– environmental conditions;
– measures against systematic failure (e.g. within generic design rules completed by elements within the risk assessment document);
– software documentation according to Clause 8;

NOTE 2 In general, this documentation is foreseen as being for the manufacturer's internal purposes and will not be distributed to the machine user.

If well-tried components are used, the documentation of these components shall include following aspects:

– version, component and application description,
– application specific information
  - use limits for the component to be regarded as well-tried,
  - suitability analysis: e.g. functional behaviour, accuracy, behaviour in the case of a fault, time response, usability and maintainability,
  - required testing,
– when based on past use for the demonstration of equivalence between the intended operation and the previous operation experience, an impact analysis on the differences between past use case and current situation shall be present.

Table 9 summarizes the documentation to be available, where appropriate.

Table 9 – Documentation of an SCS

Information required	Subclause
Functional safety plan	4.3
Safety requirements specification (SRS)	5.2
Functional requirements specification (for SCS)	5.2.3
Safety integrity requirements specification (for SCS)	5.2.5
SCS design	6.3
Structured design process	4.2
Structure of sub-functions	6.3.2
SCS architecture	6.3.2
Sub-function and Subsystem safety requirements	6.3.3
Subsystem realization	7.1
Subsystem architecture	7.2
Fault exclusions claimed when estimating fault	7.3.3.3 and 7.4.1
Subsystem assembly	7.4 and 7.5
Software safety requirements	8.3.2.2 or 8.4.2.2
Software based parameterization	6.7.5
Software configuration management items	8.3.8 or 8.4.10
Suitability of software development tools	8.3.1.3 or 8.4.1.3
Documentation of the application program	8.3.7 or 8.4.9
Results of application software module testing	8.3.5 or 8.4.6

Information required	Subclause
Results of application software integration testing	8.4.7
Documentation of SCS integration (testing)	9.5.4
Documentation of well-tried components	10.2
Documentation for installation, use and maintenance	10.3
Documentation of SCS validation	9.4 and 9.5
Documentation for SCS configuration management	4.4

Refer to Annex I which provides example of activities, documents and roles.

10.3 Information for use of the SCS

10.3.1 General

10.3.1.1 Overview

The information for use of the SCS shall provide relevant information for installation, use and maintenance. This shall include a comprehensive description of the equipment, installation and mounting as follows.

10.3.1.2 Specification of safety integrity

Specific information shall be provided on the safety integrity of the SCS, as follows:

– SIL 1, 2 or 3,
– if relevant, architectural constraints of the subsystem(s).

10.3.1.3 SCS and subsystems

SCS are typically designed and implemented as a safety-related system by a machine manufacturer using available separate subsystems.

Subsystems are typically manufactured and placed on the market as a complete device ready for use.

Therefore, there are different requirements for the information for use that apply to the manufacturer of the machine or the manufacturer of the subsystems. A manufacturer of a machine can also have the role of a manufacturer of the SCS subsystem.

10.3.2 Information for use given by the manufacturer of subsystems

The principles of ISO 12100:2010, 6.4 and the applicable sections of other relevant documents (e.g. IEC 60204-1:2016, Clause 17), shall be applied.

In particular, the manufacturer of a subsystem shall indicate in the instructions that information which is important for the safe installation, use and maintenance of the subsystem. This shall include, but is not limited to, the following:

a) description of the subsystem including:
   – general description of the subsystem and its function;
   – installation instructions;
   – interfacing requirements;
   – configuration, settings or programming information, where applicable a statement of the intended use of the subsystem and any measures that can be necessary to prevent reasonably foreseeable misuse;

b) information on operating limits of the subsystem including:
   - specification of environmental limits, e.g. temperature, lighting, vibration, noise, atmospheric contaminants;
   - specification of interfacing limits, e.g. electrical, hydraulic, pneumatic or mechanical characteristics;
   - specification of any other limits relevant to the intended safety functionality, e.g. operating frequency, strength, range;

c) a description of any fault exclusions essential for maintaining the intended safety integrity. Appropriate information (e.g. for modification, maintenance and repair) shall be given to ensure the continued justification of the fault exclusion(s);

d) a description of any necessary measures at the subsystem to ensure that there will be no degradation of the intended SCS function caused by a machine control system;

e) response time of the subsystem;

f) useful lifetime of the subsystem;

g) information on diagnostic functions required for correct interfacing and safe use;

h) information on indications and alarms;

i) the nature and frequency of any required inspection procedures;

j) the nature and frequency of any required test procedures, e.g. testing, whether the diagnostic is still working;

k) provisions for the maintainability of the subsystem where relevant. All information for maintenance shall comply with ISO 12100:2010, 6.4.5.1 e). The information shall include:
   - procedures for fault diagnosis and repair;
   - procedures for confirming correct operation subsequent to repairs;

l) safety related parameters (e.g. *PFH*, *PFD*, SIL, …).

10.3.3 Information for use given by the SCS integrator

The SCS integrator (typically the manufacturer of the machine) shall include the relevant information in the instructions for use to enable the machine user to develop procedures to ensure that the required functional safety of the SCS is maintained during use and maintenance of the machine.

In particular, SCS integrator shall indicate in the instructions that information which is important for the safe use of the SCS including information on any measures that can be necessary to prevent reasonably foreseeable misuse.

Information for use shall include, but is not limited to, the following:

a) operating limits of the SCS (including environmental conditions);

b) clear descriptions and related instructions for the user interfaces with the SCS e.g. operator panel, indications and alarms;

c) description of the safety functions implemented in the SCS, including description of hazards and hazardous situation, demand mode of operation, the safe state, process safety time, overview (block) diagram(s) and circuit diagram(s) where appropriate;

d) a description (including interconnection diagrams) of the interaction (if any) between the SCS function (s) and the machine control system function(s);

e) marking if required, according to ISO 12100:2010, 6.4.4;

f) useful lifetime and requirements for the SCS components;

g) information related to any muting and/or suspension of safety functions;

h) any operating mode relevant to the safety function(s);

i) inspection and periodic testing where relevant (e.g. certain safety distances have to be tested periodically) including the nature of any required test procedures, see also for details 6.9;

j) the tools necessary for maintenance and re-commissioning, and the procedures for maintaining the tools and equipment;

k) provisions for maintenance of the SCS where relevant, including any implications for fault exclusion. All information for maintenance shall comply with ISO 12100:2010, 6.4.5.1 e). The information shall include:

   – procedures for fault diagnosis and repair, e.g. instructions for SCS functional recovery in case of failure,

   – procedures for confirming correct operation subsequent to repairs,

   – preventive maintenance and corrective maintenance.

NOTE 1 Periodic tests are those functional tests necessary to confirm correct operation and to detect faults.

NOTE 2 Preventive maintenance are the measures taken to maintain the required performance of the SCS.

NOTE 3 Corrective maintenance includes the measures taken after the occurrence of specific fault(s) that bring the SCS back into the "as-designed state".

Annex A
(informative)

Determination of required safety integrity

A.1 General

This informative annex provides methods of qualitative approach for risk estimation and SIL assignment that can be applied for determining the required SIL of 5.2. The method as described is not intended for functions that operate in low demand mode of operation.

NOTE 1 Whenever a risk assessment has indicated a safe control measure is required, the matrix method in Clause A.2 can be used to determine the required SIL.

Experience in successfully dealing with similar machines/hazards should be taken into account when estimating the required SIL.

NOTE 2 Other risk estimation methods for specific types of machine can be used as appropriate (e.g. methods are available in IEC 61508-5 and IEC 61511-3). Therefore, the SIL required by a type-C standard can deviate from that indicated by the generic approaches given in this annex.

A.2 Matrix assignment for the required SIL

A.2.1 Hazard identification/indication

Indicate the hazards, including those from reasonably foreseeable misuse, whose risks are to be reduced by implementing an SCS. List them in the hazard column in Table A.5.

A.2.2 Risk estimation

Risk estimation should be carried out for each hazard by determining the risk parameters that as shown in Figure A.1 should be derived from the following:

– severity of harm, Se; and
– probability of occurrence of that harm, which is a function of:
  - frequency and duration of the exposure of persons to the hazard, Fr;
  - probability of occurrence of a hazardous event, Pr; and
  - possibilities to avoid or limit the harm, Av.

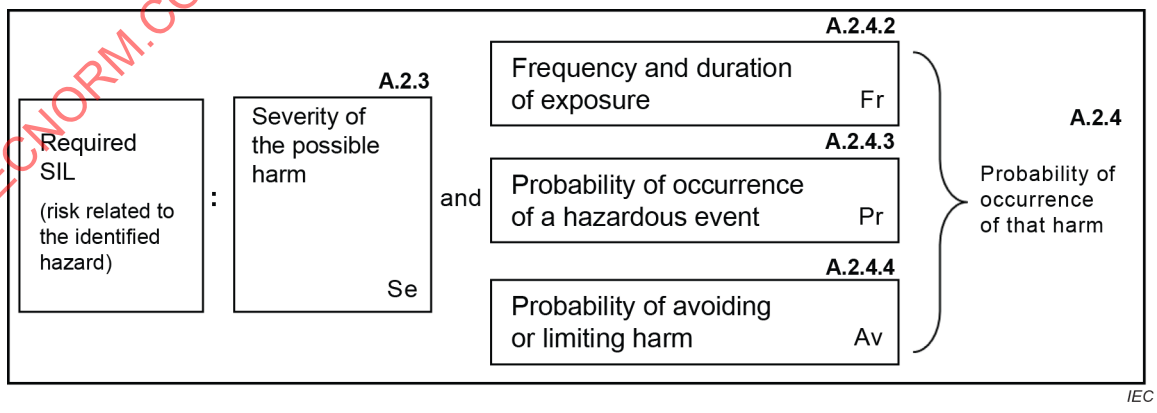


Figure A.1 – Parameters used in risk estimation

The estimates entered into Table A.5 should normally be based on worst-case considerations and need to be justified. However, in a situation where, for example, an irreversible injury is possible but at a significantly lower probability than a reversible one, then each severity level should have a separate line on the table. It can be the case that a different SCS is implemented for each line. If one SCS is implemented to cover both lines, then the highest target SIL or PL requirement should be used.

Depending on the individual application, available information such as service experience and incident statistics might be taken into account to select the ranking of the parameters.

A.2.3 Severity (Se)

Severity of injuries or damage to health can be estimated by taking into account reversible injuries, irreversible injuries and death. Choose the appropriate value of severity from Table A.1 based on the consequences of an injury, where:

4 is a fatal or a significant irreversible injury such that it will be impossible or at least very difficult to continue the same work after healing, e.g. loss of limbs, pulmonary permanent damages, loss of an eye or partial or total loss of the sight;

3 is a major or irreversible injury in such a way that it can be possible to continue the same work after healing such as loss of some fingers or toes. It can also include a severe major but reversible injury such as broken limbs;

2 is a more severe reversible injury which requires attention from a medical practitioner and it is possible to resume the work activity after a short period of time, e.g. severe lacerations, stabbing, and severe bruises;

1 is a slight injury where first aid cares without medical intervention are sufficient, e.g. minor injury including scratches and minor bruises.

NOTE For examples of severity aspects, see also appendix 5 of the EU Guidelines 2010/15/EU (RAPEX).

Select the appropriate row for consequences (Se) of Table A.1. Insert the appropriate number under the Se column in Table A.5.

Table A.1 – Severity (Se) classification

Consequences	Severity (Se)
Irreversible: death, losing an eye or arm	4
Irreversible: broken limb(s), losing a finger(s)	3
Reversible: requiring attention from a medical practitioner	2
Reversible: requiring first aid	1

A.2.4 Probability of occurrence of harm

A.2.4.1 General

Each of the three parameters of probability of occurrence of harm (i.e. Fr, Pr and Av) should be estimated independently of each other. A worst-case assumption needs to be used for each parameter to ensure that SCS(s) are not incorrectly assigned a lower PL/SIL than is necessary. Generally, the use of a form of task-based analysis is strongly recommended to ensure that proper consideration is given to estimation of the probability of occurrence of harm.

A.2.4.2 Frequency and duration of exposure

On determination of the exposure level of people to a hazard, according to 5.5.2.3.1 of ISO 12100:2010, the work situation should be assessed considering factors such as:

– the mode of operation during the access (setting/automatic/manual/special mode);

– nature of access (feeding of materials, correction of malfunction, maintenance or repair);

– time spent in the hazardous area;

– frequency of access to the hazardous area.

The parameter Fr is defined by frequency of presence of the people in the hazardous area and by the average duration of presence.

It should then be possible to estimate the interval between accesses to a hazardous area and therefore the frequency of the exposure to a potential hazard ~~(referred to a period ≥ to one year)~~. This factor does not include consideration of the failure of the SCS.

Select the appropriate row for frequency and duration of exposure (Fr) of Table A.2. Insert the appropriate number under the Fr column in Table A.5.

Table A.2 – Frequency and duration of exposure (Fr) classification

Frequency and duration of exposure (Fr)		
Frequency of exposure	**Frequency, Fr**	
	Duration of exposure ≥ 10 min	Duration of exposure < 10 min
≥ 1 per h	5	5
< 1 per h to ≥ 1 per day	5	4
< 1 per day to ≥ 1 per 2 weeks	4	3
< 1 per 2 weeks to ≥ 1 per year	3	2
< 1 per year	2	1

A.2.4.3 Probability of occurrence of a hazardous event

The probability of occurrence of harm should be estimated independently of other related parameters Fr and Av. A worst-case assumption should be used for each parameter to ensure that SCS(s) are not incorrectly assigned a lower SIL than is necessary. To prevent this occurring, the use of a form of task-based analysis is strongly recommended to ensure that proper consideration is given to estimation of the probability of occurrence of harm.

This parameter can be estimated by taking into account:

a) Predictability of the behaviour of component parts of the machine relevant to the hazard in different modes of use (e.g. normal operation, maintenance, fault finding).

   – This will necessitate careful consideration of the control system especially with regard to the risk of unexpected start up. Do not take into account the protective effect of any SCS. This is necessary in order to estimate the amount of risk that will be exposed if the SCS fails. In general terms, it shall be considered whether the machine or material being processed has the propensity to act in an unexpected manner.

   – The machine behaviour will vary from very predictable to not predictable but unexpected events cannot be discounted.

   NOTE 1 Predictability is often linked to the complexity of the machine function.

b) The specified or foreseeable characteristics of human behaviour with regard to interaction with the component parts of the machine as an origin of to the hazard. This can be characterised by:

   – stress (e.g. due to time constraints, work task, perceived damage limitation); and/or

   – lack of awareness of information relevant to the hazard. This will be influenced by factors such as skills, training, experience, and complexity of machine/process.

These attributes are not usually directly under the influence of the SCS designer, but a task analysis will reveal activities where total awareness of all issues, including unexpected outcomes, cannot be reasonably assumed.

"Very high" probability of occurrence of a hazardous event should be selected to reflect normal production constraints and worst case considerations. Positive reasons (e.g. well-defined application and knowledge of high level of user competences) are required for any lower values to be used.

Any required or assumed skills, knowledge, etc. should be stated in the information for use.

Select the appropriate row for probability of occurrence of hazardous event (Pr) of Table A.3. Indicate the appropriate number under the Pr column in Table A.3.

Table A.3 – Probability (Pr) classification

Probability of occurrence	Probability (Pr)
Very high	5
Likely	4
Possible	3
Rarely	2
Negligible	1

A.2.4.4 Probability of avoiding or limiting harm (Av)

This parameter describes whether harm could be avoided or limited in case of a hazardous event. For example, the exposure to a hazard can be directly identified by its physical characteristics, or recognized only by technical means, e.g. indicators. The probability of avoiding or limiting harm (or the controllability) is predominantly determined by human intervention and depends to a large extent on individual human abilities. Avoidance shall not be used as a substitute for basic hazard elimination. Avoiding or limiting harm considers, for example:

– characteristics of the hazardous event:
  - speed/acceleration: evolves quickly or slowly;
  - the nature of the component or system, for example a knife is usually sharp, a pipe in a dairy environment is usually hot, electricity is usually hazardous by its nature but is not visible;
  - possibility of recognition of a hazard, for example electrical hazard: a copper bar does not change its aspect whether it is under voltage or not; to recognize if one needs an instrument to establish whether electrical equipment is energised or not; ambient conditions, for example high noise levels can prevent a person hearing a machine start;
  - complexity of the operations (human interaction in terms of numbers of operation and/or timing available for these operations);
– spatial possibility to withdrawn from the hazard;
– human abilities:
  - skills of persons involved;
  - abilities to react (e.g. take action, escape, etc.);
  - aspects that reduce the ability (e.g. stress, distraction, fatigue).

NOTE Human abilities cannot be accounted more than once for each safety function.

Select the appropriate row for probability of avoidance or limiting harm (Av) of Table A.4. Insert the appropriate number under the Av column in Table A.4.

Table A.4 – Probability of avoiding or limiting harm (Av) classification

Probabilities of avoiding or limiting harm (Av)	
Impossible	5
Rarely	3
Probable	1

The classification should only be set as probable if the hazard is clearly recognizable and if there is sufficient time to take counteractions or to leave the hazardous area.

A.2.5 Class of probability of harm (Cl)

For each hazard, and as applicable, for each severity level, add up the points from the Fr, Pr and Av columns and enter the sum into the column Cl (where Cl = Fr + Pr + Av) in Table A.5.

Table A.5 – Parameters used to determine class of probability of harm (Cl)

ID.	Hazard	Fr	Pr	Av	Cl
1					
2					
3					
4					

A.2.6 SIL assignment

Using Table A.6, where the severity (Se) row crosses the relevant column (Cl), the intersection point indicates whether action is required. The black area indicates the SIL assigned as the target for the SCS. The lighter shaded areas should be used as a recommendation that other measures (OM) be used.

Where function(s) have safety implications but application leads to a required safety integrity less than that required by SIL 1 (OM or No SIL), compliance with the requirements of IEC 60204-1 or other relevant standards can lead to an adequate performance of the control system.

Table A.6 – Matrix assignment for determining the required SIL (or PL_r) for a safety function

<table>
<tr><th rowspan="2">Consequences</th><th rowspan="2">Severity Se</th><th colspan="13">Class Cl = Fr + Pr + Av</th></tr>
<tr><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td rowspan="2">Death, losing an eye or arm</td><td rowspan="2">4</td><td colspan="2">SIL 1</td><td colspan="3">SIL 2</td><td colspan="3">SIL 2</td><td colspan="3">SIL 3</td><td colspan="2">SIL 3</td></tr>
<tr><td>PL_r b</td><td>PL_r c</td><td colspan="3">PL_r d</td><td colspan="3">PL_r d</td><td colspan="3">PL_r e</td><td colspan="2">PL_r e</td></tr>
<tr><td rowspan="2">Permanent injury, losing fingers</td><td rowspan="2">3</td><td colspan="6" rowspan="2">No SIL (or PL) required</td><td colspan="3">OM</td><td colspan="3">SIL 1</td><td colspan="3">SIL 2</td><td colspan="2">SIL 3</td></tr>
<tr><td colspan="3">PL_r a</td><td>PL_r b</td><td colspan="2">PL_r c</td><td colspan="3">PL_r d</td><td colspan="2">PL_r e</td></tr>
<tr><td rowspan="2">Reversible injury, medical attention</td><td rowspan="2">2</td><td colspan="6" rowspan="2"></td><td colspan="4">OM</td><td colspan="3">SIL 1</td><td colspan="2">SIL 2</td></tr>
<tr><td colspan="4">PL_r a</td><td>PL_r b</td><td>PL_r c</td><td colspan="2">PL_r d</td></tr>
<tr><td rowspan="2">Reversible injury, first aid</td><td rowspan="2">1</td><td colspan="6" rowspan="2"></td><td colspan="5">OM</td><td colspan="2">SIL 1</td></tr>
<tr><td colspan="5">PL_r a</td><td>PL_r b</td><td>PL_r c</td></tr>
<tr><td colspan="15">EXAMPLE: For a specific hazard with an Se assigned as 3, an Fr as 4, an Pr as 5 and an Av as 5 then:

Cl = Fr + Pr + Av = 4 + 5 + 5 = 14

Using this table, this would lead to a SIL 3 or PL e being assigned to the safety function that is intended to mitigate against the specific hazard.</td></tr>
<tr><td colspan="15">NOTE 1 SIL 2 at Class 3 and 4 in the previous published edition is now reduced to SIL 1 because of the low score for the classes of Frequency, Probability and Avoiding Harm.

NOTE 2 Due to characteristics of risks present at machinery, SIL 4 is not considered in this document. For SIL 4, see IEC 61508-1.

NOTE 3 This correspondence between SIL and PL_r is valid only for the required level and not for the achieved level.

NOTE ~~3~~4 OM: Other Measures (e.g. basic safety principles, Table 7)</td></tr>
</table>

Figure A.2 shows an example of documentation.

Risk assessment and safety measures

Document No.:
Part of:

Product:
Issued by:
Date:

Black area = Safety measures required
Grey area = Safety measures recommended

Consequences	Severity	Class Cl					Frequency, Fr			Probability of hzd event, Pr	Avoidance, Av
	Se	3- 4	5 - 7	8 - 10	11 - 13	14 - 15	Duration of exposure at	≥ 10 min	< 10 min		
Death, loosing an eye or arm	4	SIL 1	SIL 2	SIL 2	SIL 3	SIL 3	≥ 1 per hr	5	5	Common	
Permanent, loosing fingers	3		OM	SIL 1	SIL 2	SIL 3	< 1 per hr ≥ 1 Per day	5	4	Likely	
Reversible, medical attention	2			OM	SIL 1	SIL 2	< 1 per day ≥ 1 Per 14 day	4	3	Possible	Impossible
Reversible, first aid	1				OM	SIL 1	< 1 per 2 wks ≥ 1 Per yr	3	2	Rarely	Possible
							≥ 1 per yr	2	1	Negligible	Likely

Ser. No.	Hzd. No.	Hazard	Se	Fr	Pr	Av	Cl	Safety measure	Safe

Comments

IEC

Figure A.2 – Example proforma for SIL assignment process

A.3 Overlapping hazards

If several hazards can be caused in a single zone by the failure of a single safety-related function, these are called overlapping hazards.

For the quantification of risk, each hazard can be evaluated separately, except when it is obvious that there is a combination of directly linked hazards which always occur simultaneously.

EXAMPLE 1 A continuous welding robot can create various simultaneous hazardous situations, for example crushing caused by movement and burning due to the welding process. This can be considered as a combination of directly linked hazards.

EXAMPLE 2 For a robot cell in which separate robots are working, each robot is considered separately.

Annex B
(informative)

Example of SCS design methodology

B.1 General

Examples of typical safety functions are cited in Table G.1. In the following example, “safety-related stopping initiated by a guard” the basic methodology of Clause 6 and Clause 7 will be shown.

This example is not intended to draw the attention of the designer on a correct mechanical design (e.g. not having a common striking plate for the two position switches) that nevertheless has to be considered by the designer of an SCS. It is intended to be a general example about how to proceed for a design of an SCS based on this document.

In the example, it is assumed that the safety function is operated in high demand mode of operation.

B.2 Safety requirements specification

The relevant information can be exemplarily summarized as shown in Table B.1.

Table B.1 – Safety requirements specification – example of overview

Functional requirements specification of the safety function	
Description:	When the guard door will be opened then the electrical motor shall stop
Condition(s):	Operating mode “all”
Restart/Reset	Manual human action required where the operator can stay in the hazard zone (according to ISO 12100:2010, 6.2.11.3)
Priority:	High priority in comparison to other safety functions; emergency stop function will have the highest priority
Frequency of operation:	10 time per hour; 16 hours per day; 250 days per year
Response time:	Maximum 500 ms from initiation event (opening of the guard door) to de-energizing electrically the self-braking motor, including the stop of the mechanical parts that are the source of the hazard (stop category 0 according to IEC 60204-1)
Interface(s) to other machine functions:	Realized by a pre-designed safety-related device (information for use of the component manufacturer to be referenced)
Fault reaction function:	Stopping immediately or detection at restart at least by prohibiting restart
Defeating:	Design of guard door and mounting of guard interlocking devices according to ISO 14119
Environment	Temperature, dust, vibration, …
Safety integrity requirements specification of the safety function	
Required SIL or PL	SIL 2 with related target PFH value (see Table 1)
Architectural constraints	– Use of mechanical guard interlocking devices (position switches) due to vibration – No type C standard requirements (e.g. required HFT)

B.3 Decomposition of the safety function

The safety function can be decomposed logically into sub-functions which can be allocated physically to subsystems, see Figure B.1.

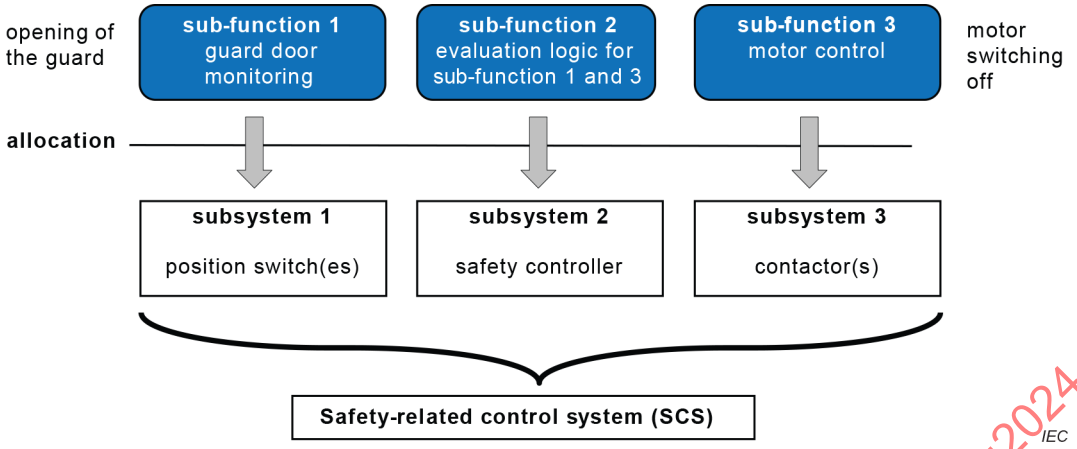


Figure B.1 – Decomposition of the safety function

B.4 Design of the SCS by using subsystems

B.4.1 General

Figure B.2 shows a technical solution for the subsystems design. For subsystem 2, the relevant safety-related data are available and are assumed as shown in Figure B.2.

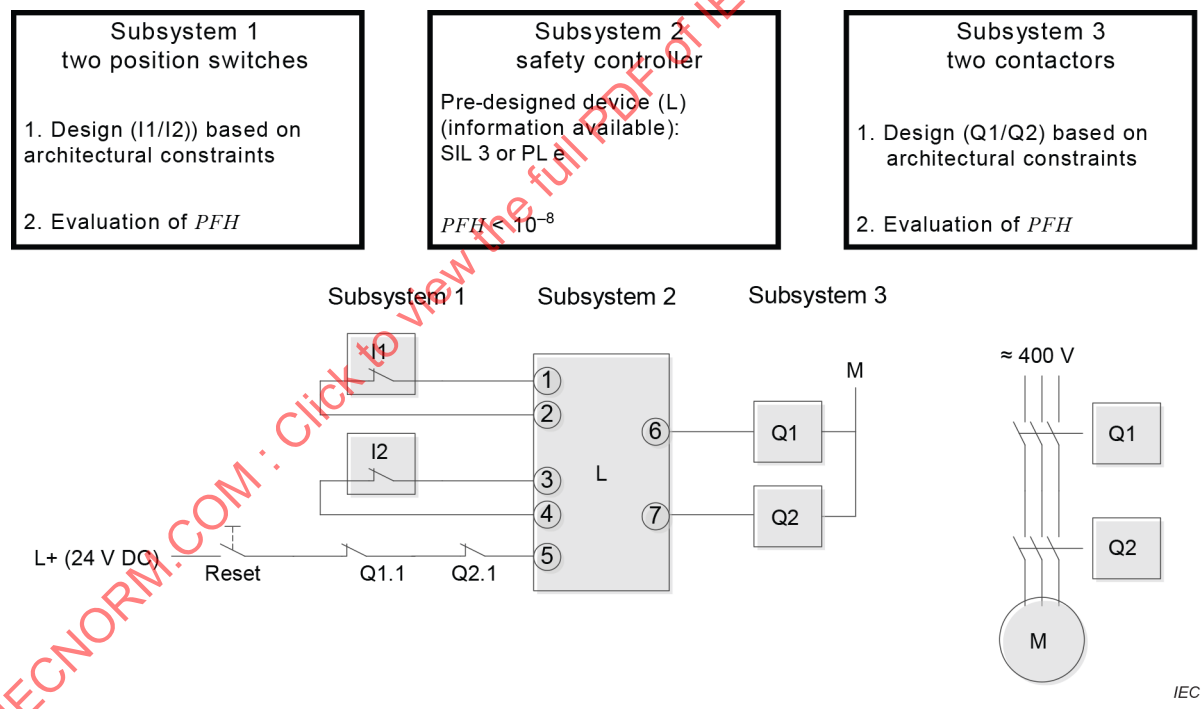


Figure B.2 – Overview of design of the subsystems of the SCS

B.4.2 Subsystem 1 design – "guard door monitoring"

B.4.2.1 Architectural constraints

This subsystem is to be designed and evaluated as described in Clause 7. Regarding a required SIL 2, an architecture with a hardware fault tolerance equal to 1 (HFT 1) has been chosen, see Table 6.

B.4.2.2 Evaluation of *SFF*

The safe failure fraction (SFF) can be calculated using the following equation:

$$SFF = \frac{\sum \lambda_S + \sum \lambda_{DD}}{\sum \lambda_S + \sum \lambda_D} \quad \text{(B.1)}$$

where

λ_S is the rate of safe failure,

$\sum \lambda_S + \sum \lambda_D$ is the overall failure rate,

λ_{DD} is the rate of dangerous failure which is detected by the diagnostic functions;

λ_D is the rate of dangerous failure.

Depending from the safety function, a failure can be safe (λ_S) or dangerous (λ_D).

Identification of failure modes:

A fault of an electromechanical component generally represents a situation (state) that can lead to a failure. Assuming that the safe state is an open circuit:

– the contact remains open: safe state;

– the contact remains closed: dangerous state.

The theoretical failure effects of the position switch are:

– the contact will not (anymore) open: dangerous failure (unintended closed);

– the contact will open "by itself": safe failure (unintended opened, can be considered as very unlikely for an electromechanical device);

– the contact will not (anymore) close: safe failure which do not have any influence of the safety function (unintended opened);

– the contact will close "by itself": dangerous failure (unintended closed).

NOTE See also failure modes in IEC 60947-4-1.

Practical considerations:

The opening of the guard door defines the failure modes of the position switch to be considered. That means that practically no safe failures of the position switch related to this safety function exist:

– the failure mode “unintended closed” contact is always dangerous (typical dangerous failure of the position switch);

– the failure mode “unintended opened“ contact is not relevant for the opening of the guard door and only has an influence on the availability of the machine. It is a no effect failure (IEC 61508-4:2010, 3.6.14) for the defined function. Therefore, it is not a safe failure and λ_S = 0.

Evaluation of *SFF*:

The safe failure fraction in this example is given by following equation:

$$SFF = \frac{\lambda_{DDI1} + \lambda_{DDI2}}{\lambda_{DI1} + \lambda_{DI2}} = \frac{\lambda_{DDI1}}{\lambda_{DI1}} = \frac{\lambda_{DDI2}}{\lambda_{DI2}} = DC_{I1} = DC_{I2} \quad \text{(B.2)}$$

where

– λ_S = 0 is used;

– the fundamental definition of $\lambda_{DD} = DC \times \lambda_D$ is used;

– I1 and I2 have the same failure rates;

– DC_{I1} of I1 and DC_{I2} of I2 are equal due to cross-checking.

B.4.2.3 Evaluation of DC_{I1} and DC_{I2}

DC of 99 % can be assumed based on Table D.1:

– "Cross monitoring of input signals and intermediate results within the logic (L) and temporal and logical software monitor of the program flow and detection of static faults and short circuits (for multiple I/O)".

According to Table 6, the subsystem can claim maximum SIL 3.

B.4.2.4 Evaluation of PFH

B.4.2.4.1 Failure rates of position switches (I1/I2)

The failure rate will be determined by using B_{10D} (or B_{10} and RDF) as follows (see 7.3.4.2):

– $n_{op} = \dfrac{d_{op} \times h_{op} \times 3\,600 \frac{s}{h}}{t_{cycle}} = \dfrac{250\ d \times 16 \frac{h}{d} \times 3\,600 \frac{s}{h}}{360\ s} = 40\,000$ cycles per year;

– $MTTF_D = \dfrac{B_{10D}}{0{,}1 \times n_{op}} = \dfrac{2\,000\,000}{0{,}1 \times 40\,000} = 500$ years;

– $\lambda_D = \dfrac{1}{MTTF_D \times 8\,760 \frac{h}{a}} = 2{,}28 \times 10^{-7}$ failures per hour.

NOTE In this example, B_{10D} of Table C.1 (position switches with separate actuator) is used. In practice, the safety-related data provided by the component manufacturer will be used.

B.4.2.4.2 Annex H approaches

– Table allocation approach (see Table H.1): PFH = 0,10 x 10^{-7}. This approach is simple but gives a conservative approach.

– Formulas (see H.2.5): PFH = 4,65 × 10^{-9} (with β= 2 %, T_1 = 175 200 h, T_2 = 1/C = ~~n_{op}/8 760 h~~ 8 760 h / n_{op}). This approach is more complex but gives a more accurate result.

B.4.3 Subsystem 2 design – "evaluation logic"

According to the data provided by the component manufacturer of the pre-designed safety controller, subsystem 2 can claim SIL 3 with a PFH < 10^{-8}.

NOTE The instruction for use of the component manufacturer will provide all safety-related information.

B.4.4 Subsystem 3 design – "motor control"

B.4.4.1 Architectural constraints

B.4.4.1.1 General

This subsystem is to be designed and evaluated as described in Clause 7. Regarding a required SIL 2, an architecture with a hardware fault tolerance equal to 1 (HFT = 1) has been chosen, see Table 6.

B.4.4.1.2 Evaluation of SFF

The same approach as described in B.4.2.2 is valid: $SFF = DC_{Q1} = DC_{Q2}$.

B.4.4.1.3 Evaluation of DC_{Q1} and DC_{Q2}

The diagnostic function will disable a reset when the mirror contacts (Q1.1 and Q2.1, Figure B.2) are not closed.

DC of 99 % can be assumed based on Table D.1:

– "Redundant shut-off path with monitoring of the actuators by logic and test equipment".

According to Table 6, the subsystem can claim maximum SIL 3.

B.4.4.2 Evaluation of PFH

B.4.4.2.1 Failure rates of contactors (Q1/Q2)

The failure rate will be determined by using B_{10D} (or B_{10} and RDF) as follows (see 7.3.4.2):

– $n_{op} = \dfrac{d_{op} \times h_{op} \times 3\,600 \frac{s}{h}}{t_{cycle}} = \dfrac{250\ d \times 16 \frac{h}{d} \times 3\,600 \frac{s}{h}}{360\ s} = 40\,000$ cycles per year ;

– $MTTF_D = \dfrac{B_{10D}}{0{,}1 \times n_{op}} = \dfrac{1\,300\,000}{0{,}1 \times 40\,000} = 325$ years;

– $\lambda_D = \dfrac{1}{MTTF_D \times 8\,760 \frac{h}{a}} = 3{,}51 \times 10^{-7}$ failures per hour .

NOTE In this example, B_{10D} of Table C.1 (contactors with nominal load) is used. In practice, the safety-related data provided by the component manufacturer will be used.

B.4.4.2.2 Annex H approaches

– Table allocation approach (see Table H.1): PFH = 0,10 x 10^{-7}. This approach is simple but gives a conservative result.

– Formulas (see H.2.5): PFH = 7,23 x 10^{-9} (with β = 2 %, T_1 = 175 200 h, T_2 = 1/C = ~~n_{op}/8 760 h~~ 8 760 h / n_{op}). This approach is more complex but gives a more accurate result.

B.4.5 Evaluation of the SCS

B.4.5.1 Target

The SCS can reach SIL 3 (see ~~6.4.2~~ 6.4.1).

B.4.5.2 Systematic integrity and CCF

The relevant requirements for each subsystem design are given in 7.3.2. Table B.2 gives an overview. The evaluation of the common cause failures (see Annex E) is based on the measures of the systematic integrity and on the architecture of the SCS.

B.4.5.3 Architectural constraints

All subsystems are claiming SIL 3. This SCS can reach SIL 3 (see ~~6.4.2~~ 6.4.1).

Table B.2 – Systematic integrity – example of overview

Avoidance of systematic failures	
Measure	**Comment/Examples**
Proper selection, combination, arrangements, assembly	The right component for the application and used in the correct way (instruction for use of the component manufacturer): Position switches useable for this application?
Wiring, interconnections	See instruction for use of the component manufacturer; measures against short-circuit failures
Correct dimensioning and shaping	Electrical overdimension: Load of contactors correct?
Hardware design review	Analysis of plausibility and considering the instruction for use of the component manufacturer
Control of systematic failures	
Measure	**Comment/Examples**
Voltage variations and interruptions	Hardware design according to IEC 60204-1
Effects of the physical environment and EM immunity	Hardware design according to IEC 60204-1; see instruction for use of the component manufacturer
Effects of temperature increase or decrease	See instruction for use of the component manufacturer; if necessary including relevant hints into the information for use of the safety function
Use of de-energization	Switching off of the motor by actuating the position switches
Well-tried safety principles	
– Failure detection by automatic tests	Monitoring of two position switches and two contactors; short-circuit detection
– Operation in the positive mode	Positively driven contacts of the position switches
– Mechanically linked contacts	Contactors with mirror contacts

B.4.6 PFH

The overall PFH by summation of the PFH of the three subsystems will be < 10^{-7}.

This SCS reaches SIL 3 (see 6.4.2).

B.5 Verification

B.5.1 General

The overall validation process requires at each design and evaluation state different verification activities (see validation principles represented in Figure 15).

B.5.2 Analysis

Check of plausibility of the safety requirements specification (see Clause B.2), the decomposition of the safety function (see Clause B.3) and the design and evaluation of the SCS (see Clause B.4).

B.5.3 Tests

The following tests of Table B.3 should be performed.

Table B.3 – Verification by tests

Tests	Examples of test criteria
Testing by functional tests of the SCS	– Opening of the guard door and verifying of the stop of the motor – Closing of the guard door and verifying that no restart occurs
Tests by using fault injection • Subsystem “guard door monitoring”: • Subsystem “evaluation logic”: • Subsystem “motor control”:	Verification of detection of discrepancy between the position switch input signals (e.g. by dismounting of one of the positions switches and short-circuit test) Test of reset when the guard door is opened and test when a failure is present (during the test by using fault injection) Verification of monitoring of the mirror contacts of the contactors (e.g. by electrical disconnection of the feedback signal of the contactors)

Annex C
(informative)

Examples of $MTTF_D$ values for single components

C.1 General

This annex describes different methods to calculate or evaluate $MTTF_D$ values for single components. Clause C.2 describes a method based on the respect of good engineering practices for different kinds of components, Clause C.3 describes a method for hydraulic components, Clause C.4 describes the calculation of $MTTF_D$ for components from B_{10}.

C.2 Good engineering practices method

If all following requirements are fulfilled, the $MTTF_D$ or B_{10D} value for a component can be estimated according to Table C.1:

a) The manufacturer of the component confirms the use of basic and well-tried safety principles according to ISO 13849-2, or the relevant standard (see Table C.1) for the design of the component (confirmation in the data sheet of the component).

b) The manufacturer of the component specifies the appropriate application and operating conditions for the subsystem designer.

c) The design of the subsystem fulfils the basic and well-tried safety principles according to ISO 13849-2, for the implementation and operation of the component.

C.3 Hydraulic components

If all following requirements are fulfilled, the $MTTF_D$ value for a single hydraulic component, e.g. valve, can be estimated as 150 years:

a) The manufacturer of the hydraulic component specifies the appropriate application and operating conditions for the subsystem designer, and

b) The manufacturer of the hydraulic component specifies the appropriate application and operating conditions for the user. The user has to be informed about his responsibility to fulfil the basic and well-tried safety principles according to ISO 13849-2 for the implementation and operation of the hydraulic component.

If the criteria presented in Clause C.4 are met, the $MTTF_D$ value for a single hydraulic component, e.g. valve, can be estimated at 150 years. If the mean number of annual operations (n_{op}) is below 1 000 000, then the $MTTF_D$ value can be estimated higher as shown in Table C.1.

But if either a) or b) is not achieved, the $MTTF_D$ value for the single hydraulic component has to be given by the manufacturer. Instead of using a fixed value for the $MTTF_D$ as described above, it is permissible to use the B_{10D} concept for $MTTF_D$ of pneumatic, mechanical and electromechanical components also for hydraulic components if the manufacturer can provide data.

C.4 $MTTF_D$ of pneumatic, mechanical and electromechanical components

For pneumatic, mechanical and electromechanical components (pneumatic valves, relays, contactors, position switches, cams of position switches, etc.), it can be difficult to calculate the mean time to dangerous failure ($MTTF_D$ for components), which is given in years and which is required by this document. Most of the time, the manufacturers of these kinds of components only give the mean number of cycles until 10 % of the components fail dangerously (B_{10D}). This clause gives a method for calculating an $MTTF_D$ for components by using B_{10} or T (lifetime) given by the manufacturer related closely to the application dependent cycles.

For relationship of relevant parameters, see 7.3.4.2.

Table C.1 – Standards references and $MTTF_D$ or B_{10D} values for components

	Basic and well-tried safety principles (ISO 13849-2:2012)	Other relevant standards	Typical $MTTF_D$ (a) or B_{10D} (cycle) values
Mechanical components	Tables A.1 and A.2		$MTTF_D$ = 150
Hydraulic components with n_{op} ≥ 1 000 000	Tables C.1 and C.2	ISO 4413	$MTTF_D$ = 150
Hydraulic components 1 000 000 > n_{op} ≥ 500 000	Tables C.1 and C.2	ISO 4413	$MTTF_D$ = 300
Hydraulic components 500 000 > n_{op} ≥ 250 000	Tables C.1 and C.2	ISO 4413	$MTTF_D$ = 600
Hydraulic components 250 000 > n_{op}	Tables C.1 and C.2	ISO 4413	$MTTF_D$ = 1 200
Pneumatic components	Tables B.1 and B.2	ISO 4414	B_{10D} = 20 000 000
Relays and contactor relays with small load (mechanical load)	Tables D.1 and D.2	EN 50205, IEC 61810, IEC 60947	B_{10D} = 20 000 000
Relays and contactor relays with maximum load	Tables D.1 and D.2	EN 50205, IEC 61810, IEC 60947	B_{10D} = 400 000
Proximity switches with small load (mechanical load)	Tables D.1 and D.2	IEC 60947, ISO 14119	B_{10D} = 20 000 000
Proximity switches with nominal load	Tables D.1 and D.2	IEC 60947, ISO 14119	B_{10D} = 400 000
Contactors with small load (mechanical load)	Tables D.1 and D.2	IEC 60947	B_{10D} = 20 000 000
Contactors with nominal load	Tables D.1 and D.2	IEC 60947	B_{10D} = 1 300 000 (see Note 1)
Position switches [a]	Tables D.1 and D.2	IEC 60947, ISO 14119	B_{10D} = 20 000 000
Position switches (with separate actuator, guard-locking) [a]	Tables D.1 and D.2	IEC 60947, ISO 14119	B_{10D} = 2 000 000
Emergency stop devices [a]	Tables D.1 and D.2	IEC 60947, ISO 13850	B_{10D} = 100 000
Push buttons (e.g. enabling switches)	Tables D.1 and D.2	IEC 60947	B_{10D} = 100 000
For the definition and use of B_{10D}, see 7.3.4.2.			

NOTE 1 B_{10D} is estimated as two times B_{10} (50 % dangerous failure) if no other information (e.g. product standard or results of analysis) is available.

NOTE 2 Small load means e.g. 20 % of the rated value. For more information, see ISO 13849-2.

NOTE 3 Emergency stop devices according to IEC 60947-5-5 and ISO 13850 and enabling switches according to IEC 60947-5-8 can be considered as a HFT = 0 or HTF = 1 subsystem depending on the number of electrical output contacts and on the fault detection ~~in the subsequent~~ provided by another subsystem of the SCS. Each contact element (including the mechanical actuation) can be considered as one channel with a respective B_{10D} value.

For enabling switches according to IEC 60947-5-8, this implies the opening function by pushing through or by releasing. In some cases, it can be possible that the machine builder can apply fault exclusion according to according to tables in Annexes A to D of ISO 13849-2:2012, considering the specific application and environmental conditions of the device.

NOTE 4 The $MTTF_D$ does not apply to systematic measures (e.g. housing).

[a] If fault exclusion for direct opening action is possible.

Annex D
(informative)

Examples for diagnostic coverage (*DC*)

Typical examples of *DC* values are given in Table D.1.

For measures where a *DC* range is given (e.g. fault detection by the process), the correct *DC* value can be determined by considering all dangerous failures and then deciding which of them are detected by the *DC* measure. In case of any doubt, a FMEA should be the basis for the estimation of the *DC*.

NOTE 1 Additional estimations for DC see e.g. IEC 61508-2:2010, Tables A.2 to A.15.

Table D.1 – Estimates for diagnostic coverage (*DC*) (1 of 2)

Measure	Diagnostic coverage (*DC*)	Examples
Input device		
Cyclic test stimulus by dynamic change of the input signals	90 %	Automatically changing an output to check whether the input connected with this output will change state
Plausibility check, e.g. use of normally open and normally closed mechanically linked contacts	99 %	Compare automatically a normally closed contact with a normally open contact off a single sensor
Cross monitoring of inputs without dynamic test	0 % to 99 %, depending on how often a signal change is done by the application	Emergency stop dual channel without short circuit detection → *DC* = 90 % Emergency stop dual channel with short circuit detection → *DC* = 99 %
Cross monitoring of input signals with dynamic test if short circuits are not detectable (for multiple I/O)	90 %	Comparing the signals of 2 position monitoring devices by realising the dynamic test through automatically moving the devices between the two positions and thus the expected position can be compared with effective position (without short circuit detection)
Cross monitoring of input signals and intermediate results within the logic (L) and temporal and logical software monitor of the program flow and detection of static faults and short circuits (for multiple I/O)	99 %	Electronic devices continuously checking its functioning. Typically, this measure is used in complex electronics.
Indirect monitoring (e.g. monitoring by pressure switch, electrical position monitoring of actuators)	90 % to 99 %, depending on the application	Monitoring a cylinder is in its end position and remains in this end position Care should be taken the system can detect a failure before a dangerous situation can exist (e.g. if the cylinder leaves its position, it should be possible to place the system automatically in a safe state)
Direct monitoring (e.g. electrical position monitoring of control valves, monitoring of electro-mechanical devices by mechanically linked contact elements)	99 %	Monitoring the functioning of a contactor by a mechanically linked NC contact
Fault detection by the process	0 % to 99 %, depending on the application; this measure alone is not sufficient for SIL 3	– Degradation of the outcome of the production process indicates a probable future loss of the safety function – A measured value (e.g. level) does not correspond with the expected value
Monitoring some characteristics of the sensor (response time, range of analogue signals, e.g. electrical resistance, capacitance)	60 %	Checking an analogue value remains within predefined borders (e.g. 12 mA to 17 mA)

Table D.1 *(2 of 2)*

Measure	Diagnostic coverage (*DC*)	Examples
Output device		
Monitoring of outputs by one channel without dynamic test	0 % to 99 % depending on how often a signal change is done by the application	Monitoring the position of a moving ~~cart~~ part or final element and thus checking the output (*DC* = 0 % if change is done less than once a year and *DC* = 90 % if change is done weekly)
Cross monitoring of outputs without dynamic test	0 % to 99 % depending on how often a signal change is done by the application	
Cross monitoring of output signals with dynamic test without detection of short circuits (for multiple I/O)	90 %	Check if the two 3/2 exhaust valves have switched off by making use of a pressure switch and switching on the valves one by one to see if a difference in pressure occurs
Cross monitoring of output signals and intermediate results within the logic (L) and temporal and logical software monitor of the program flow and detection of static faults and short circuits (for multiple I/O)	99 %	Measuring the speed after activation of Safely Limited Speed which is compared with the expected program values
Redundant shut-off path with monitoring of the actuators by logic and test equipment	99 %	Monitoring both NC mechanically linked contacts (placed in series or ~~in parallel on~~ on two separate inputs of the logic) of a redundant contactor arrangement
Indirect monitoring (e.g. monitoring by pressure switch, electrical position monitoring of actuators)	90 % to 99 %, depending on the application	Monitoring a cylinder is in its end position and remains in this end position. Care should be taken the system can detect a failure before a dangerous situation can exist (e.g. if the cylinder leaves its position, it should be possible to place the system automatically in a safe state)
Fault detection by the process	0 % to 99 %, depending on the application, this measure alone is not sufficient for the required SIL 3!	– Degradation of the outcome of the production process indicates a probable future loss of the safety function – A measured value (e.g. level) does not correspond with the expected value
Direct monitoring (e.g. electrical position monitoring of control valves, monitoring of electromechanical devices by mechanically linked contact elements)	99 %	Monitoring the functioning of a contactor by a mechanically linked NC contact

For the application of Table D.1, see the indicative example below.

EXAMPLE The *DC* measure “fault detection by the process” is only to be applied if the safety-related component is involved in the production process, e.g. a PLC or sensors are used for workpiece processing and as part of one of two redundant functional channels executing the safety function. The appropriate *DC* level depends on the overlap of the commonly used resources (logic, inputs/outputs etc.). E.g. when all faults of a rotary encoder on a printing machine lead to highly visible interruption of the printing process, the *DC* for this sensor used to monitor a safely limited speed is estimated as 90 % up to 99 %.

Annex E
(informative)

Methodology for the estimation of susceptibility to common cause failures (CCF)

E.1 General

This informative annex provides ~~two~~ a simple qualitative approach~~es~~ for the estimation of CCF that can be applied to the subsystem design.

E.2 Methodology

E.2.1 Requirements for CCF

A comprehensive procedure for measures against CCF for sensors/actuators and separately for control logic is given, for example, in IEC 61508-6:2010, Annex D. Not all measures given therein are applicable to the machinery application. The most important measures are given here.

NOTE It is assumed that for redundant systems a β-factor according to IEC 61508-6:2010, Annex D is less than or equal to 2 %.

E.2.2 Estimation of effect of CCF

This quantitative process should be passed for the whole system. Every part of the ~~safety-related parts of the control system~~ SCS should be considered.

Table E.1 lists the measures and contains associated values, based on engineering judgement, which represent the contribution each measure makes in the reduction of common cause failures. 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.

Table E.1 – ~~Criteria for estimation of CCF~~ Estimation of CCF factor *(β)*

Item	Reference	Score
Separation/segregation		
Are SCS signal cables for the individual channels routed separately from other channels at all positions? For example: – Signal cables for the individual channels separate from other channels at all positions or sufficiently shielded (connected to protective earth) – Short circuit detection provided – Sufficient clearances and creepage distances on printed-circuit boards	1a	5
Where information encoding/decoding is used, is it sufficient for the detection of signal transmission errors?	1b	10
Are SCS signals and power cables / sources separate at all positions or sufficiently shielded (no interference from any other electrical system to the SCS signals, see IEC 60204-1:2016, Annex H)?	2	5
If subsystem elements can contribute to a CCF, are they provided as physically separate devices in their local enclosures?	3	5
Diversity/redundancy		
Does the subsystem employ different technologies, for example one electronic or programmable electronic and the other an electromechanical relay or a hydraulic valve?	4	8
Does the subsystem employ elements that use different physical principles (e.g. sensing elements at a guard door that use mechanical and magnetic sensing techniques)?	5	10
Does the subsystem employ elements with temporal differences in functional operation and/or failure modes?	6	10
Do the subsystem elements have a diagnostic test interval of ≤1 min?	7	10
Complexity/design/application		
Is cross-connection between channels of the subsystem prevented with the exception of that used for diagnostic testing purposes?	8	2
Assessment/analysis		
Has an analysis been conducted to establish sources of common cause failure and have predetermined sources of common cause failure been eliminated by design? For example, over voltage, over temperature, over pressure etc. (see 7.3.2.3)	9	9
Are field failures analysed with feedback into the design?	10	9
Competence/training		
Do subsystem designers understand the causes and consequences of common cause failures?	11	4
Environmental control		
Are the subsystem elements likely to operate always within the range of temperature, humidity, corrosion, dust, vibration, etc. over which it has been tested, without the use of external environmental control? (See IEC 60068 (all parts))	12	9
Is the subsystem immune to adverse influences from electromagnetic interference (See IEC 61326-3-1 or IEC 61000-6-7)	13	9
NOTE 1 An alternative item (e.g. references 1a and 1b) is given where it is intended that a claim can be made for a contribution towards avoidance of CCF from only the most relevant item. NOTE 2 Similar criteria can be derived for other technologies based on the same principles.		

Using Table E.1, those items that are considered to affect the subsystem design should be added to provide an overall score for the design that is to be implemented. Where it can be shown that equivalent means of avoiding of CCF can be achieved through the use of specific design measures (e.g. the use of opto-isolated devices rather than shielded cables), then the relevant score can be claimed as this can be considered to provide the same contribution to the avoidance of CCF.

It is expected that the references 9, 11, 12 and 13 are always addressed unless it can be justified.

This overall score can be used to determine a common cause failure factor (β) using Table E.2.

Table E.2 – Criteria for estimation of CCF

Overall score	Common cause failure factor (β)
≤ 35	10 % (0,1)
36 to 65	5 % (0,05)
66 to 85	2 % (0,02)
86 to 100	1 % (0,01)

Annex F
(informative)

Guideline for software level 1

F.1 Software safety requirements

Relevant input and output information are given in Table F.1.

Based on one example, some guidance related to some relevant documents will be shown.

Table F.1 – Example of relevant documents related to the simplified V-model

Document	Comments
Coding guidelines	Generally reusable, see Table F.2
Specification of the safety functions	Should already exist, see Table F.3 and Figure F.2
Specification of the hardware design (see Note 1)	
– Plant sketch(s)	Should already exist, see Figure F.1
– Control system design	Should already exist (not shown in this example)
– Wiring diagram(s)	Should already exist (not shown in this example)
– I/O-list	Should already exist, see Table F.4
Software design specification (see Note 2)	
– Safety-related software specification and validation plan	Table F.8 or relevant documents clearly stating the activities
– Architecture of safety-related program	Unnecessary in case of a simple application (not shown in this example)
– Architecture of non-safety-related program	Unnecessary in case of a simple application (not shown in this example)
– Module architecture of safety-related program	Optional, see Figure F.2
– Program sketch (logical representation)	Optional, see Figure F.3
Protocols	
– Software verification	See Table F.6
– Code review	See Table F.7
– Software validation	See Table F.8
NOTE 1 Hardware printout generated by CAD tools can be used.	
NOTE 2 Software printout generated by pre-designed software-platform can be used.	

All documents should be clearly identified to ensure the interrelationship between hardware and software design:

– Date: YYYY-MM-DD (currently valid version/changes)
– Name: (responsible person)
– Software signature: (number or string, easy to track and trace, e.g. CRC value)
– Hardware signature: (number or string, easy to track and trace)

F.2 Coding guidelines

Table F.2 shows a non-exhaustive template providing a typical list of coding guidelines for SW level 1 applications. For clarification, it is populated with specific examples.

Table F.2 – Examples of coding guidelines

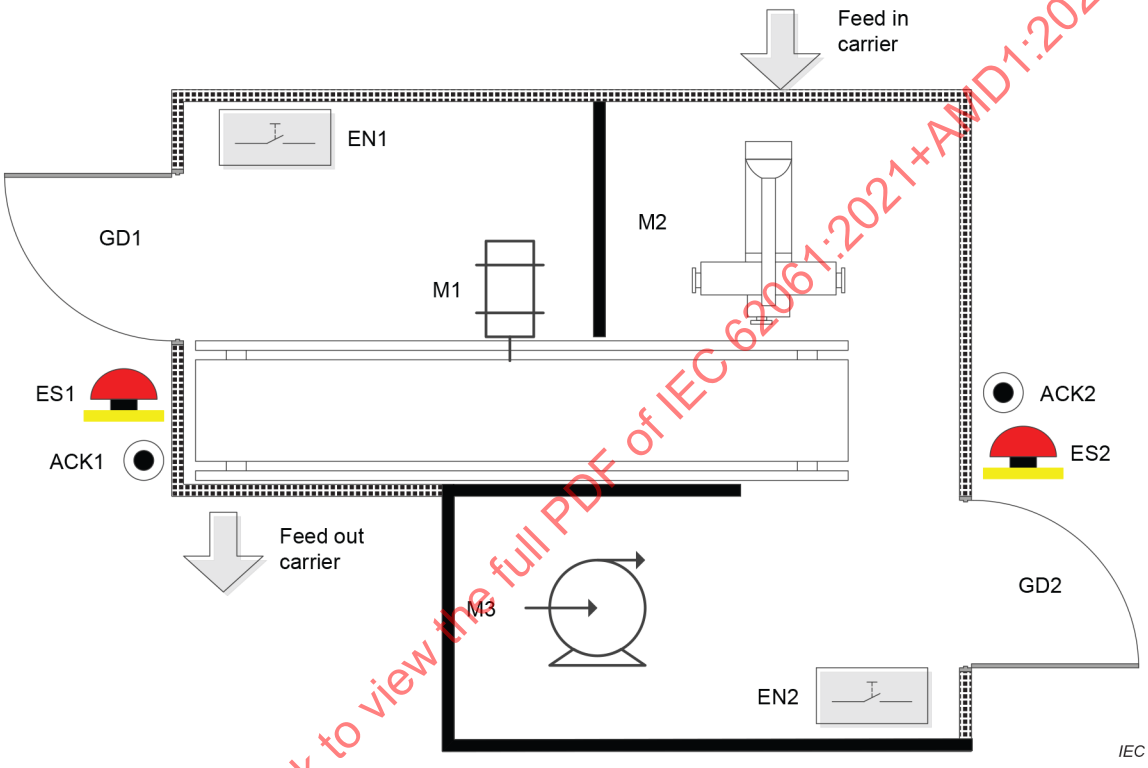
A. Variables
Prefixes of boolean variables: "b".
Prefixes of binary inputs: "I_b" (non-safety-related input), **"IS_b"** (safety-related input).
Prefixes of binary outputs: "Q_b" (non-safety-related output) or **"QS_b"** (safety-related input).
Prefixes of instances: Timers: "T_", positive edge detections: **"R_"**, Flip-Flops: **"FF_"**
Prefixes of instances: Instances of SF_GUARD: GUARD_<*guard name*>, SF_ESTOP: ESTOP_<*number*>, SF_FDBACK: CONTACTORS_<*contactors*>
Prefixes of global variables: "G_" (non-safety-related), **"GS_"** (safety).
Prefixes of temporary variables: "#"
Variable names: The variable name after the prefix should be self-explanatory, e.g. should contain the device name under consideration. For example, GD1.. for guard door 1.
Variable declaration: Initialize with the safest condition. Include a comment in each declaration.
B. Signal processing
Software architecture: Partition the software data flow in a pre-processing layer (inputs), a switch off logic (logic) and a post-processing layer (outputs). Realize the pre-processing layer in consecutive networks. The output of each network should somehow contribute to the switch off logic. For each binary output: Realize the corresponding switch off logic and the post-processing layer in one network (if possible).
Assignment: Use outputs and variables in only one program statement.
Comments: Each network has a comment.
Cyclic processing: Run each part of the safety-related software unconditionally as part of each cycle.
Monitoring of two channel inputs: Monitor on two channel inputs (e.g. push buttons) by the input cards with a discrepancy time of e.g. 100 ms.
Monitoring of contactors: Monitor of the mirror contacts of contactors with a feedback time of e.g. 1 s.
Monitoring of guard door: Monitor of the interlocking devices with a discrepancy time of e.g. 100 ms to 500 ms.
Automatic restart: Is only allowed for guard doors where the operator cannot stay in the hazard zone.
Errors in peripheral devices: Manual reset is necessary.
Triggering of safety functions: Trigger by FALSE.
Concept of acknowledge of detected failures: Selectivity of "reset/acknowledge" depending on the availability concept, human actions requirements
Response time (typical): Calculate or test and document the response time of the safety-related program.
C. Library function blocks / functions (FBs/FCs)
Usage: Wherever applicable use pre-designed library FBs/FCs.
Guard door: SF_GUARD.
Emergency stop device: SF_ESTOP.
Contactor: SF_FDBACK.
Enabling device: SF_EV2DI
Automatic Reset: Depending on the library functions (to be cited here)
Activation: Depending on the library functions (to be cited here)
Self-developed FBs/FCs: If applicable, capsule logical signal combinations which have multiple assignments within the project in a FB/FC. The life cycle complies with the V-model. These FBs/FCs shall be password protected. A library management is necessary.

F.3 Specification of safety functions

This example refers to Clause 8, especially to 8.3.1, SW level 1, and is based on the simplified V-model of Figure 12.

This example is not intended to draw the attention of the designer on a correct mechanical design (e.g. not having a common striking plate for the two position switches) that nevertheless has to be considered by the designer of an SCS. It is intended to be a general example about how to proceed for a design of the SW level 1 and is based on the simplified V-model.

The plant sketch in Figure F.1 helps to understand the safety concept.



Key

ACK1, ACK2	acknowledge (related to guard doors and emergency stop devices)
ES1, ES2	emergency stop devices (with two positively driven contacts)
EN1, EN2	enabling devices for safely limited speed
GD1, GD2	guard doors (monitored by two position switches)
M1, M2	motors controlled by frequency converters with safety-related sub-functions (STO and SLS)
M3	motor of pump (switched off by two contactors)

Figure F.1 – Plant sketch

During risk assessment, the following safety functions with a required SIL 2 or PL d are specified and summarized in Table F.3.

Table F.3 – Specified safety functions

Safety function		Functional description			Operating mode
Nr.	Description	IF (CAUSE)	THEN (EFFECT)	Response time (see Note 3)	
SF1	Emergency stop functions (see Note 1 and Note 2)	ES1 is pushed (#bES1_OK = 0)	M1 shall stop (#bM1_STO = 0)	< 1 s	all
SF2		ES2 is pushed (#bES2_OK = 0)	M1, M2 and M3 shall stop (#bM1_STO = 0) (#bM2_STO = 0) (#bM3_ON = 0)	< 1 s	all
SF10	Monitoring of guard doors	GD1 is opened (#bST1_OK = 0)	M1 shall stop (#bM1_STO = 0)	< 1 s	all
SF11		GD2 is opened (#bST2_OK = 0)	M1, M2 and M3 shall stop (#bM1_STO = 0) (#bM2_STO = 0) (#bM3_ON = 0)	< 1 s	all
SF20	Reduced speed control by using enabling devices EN1 or EN2	GD1 is opened and EN1 is pushed (#bEN1_OK = 1)	reduced speed is allowed (#bM1_STO = 1) and (#bM1_SLS = 0)	< 500 ms	reduced speed
SF21		GD2 is opened and EN2 is pushed (#bEN2_OK = 1)	reduced speed is allowed (#bM2_STO = 1) and (#bM2_SLS = 0)	< 500 ms	reduced speed

NOTE 1 An emergency stop function represents a complementary measure (according to ISO 12100). The evaluation can be made by using the principles of the design of a safety function..

NOTE 2 Depending on the area of the hazard zone, SF2 can be subdivided into several independent safety functions (see overlapping hazards, Clause A.3).

NOTE 3 The overall response time that is accepted to reach safe state.

The normal operation mode related to plant sketch in Figure F.1 is as follows:

– The maximum time of 500 ms from initiation event (opening of the guard door or activation of EN1 or EN 2) to de-energizing electrical self-braking motors and stop the mechanical parts that are the source of the hazard before they can be reached (stop category 0 according to IEC 60204-1) is accepted.
– It is not possible for an operator to pass from dangerous zone 1 to dangerous zone 2 and vice versa.
– Pieces are transported by a feed in carrier to the machine and, after the process, these will be moved by a feed out carrier. These carriers are not considered in this example.
– Milling centre (M2) is working automatically and treated pieces are transported by the carrier (M1) for cooling. The guard doors shall be closed at this time.
– Sometimes a worker opens the guard door GD2 and withdraws the piece and cleans the tool; after this, the worker goes out and closes GD2 again. After acknowledging (ACK2), the process can be restarted.
– Sometimes a worker needs to readjust the milling centre (M2) and is using therefore the enabling device EN2 to activate the reduced speed of M2. Only while the GD2 is opened this work is allowed.
– Sometimes the carrier shall be cleaned using a reduced speed. When the guard door GD1 will be opened by the worker, the carrier shall stop. Only while the GD1 is opened, GD2 is closed and the enabling device EN1 is activated, the carrier can be moved slowly for cleaning. After closing GD1, the process can be restarted by acknowledging (ACK1).

F.4 Specification of hardware design

The relevant components for the hardware design of the control system e.g. are:

– safety-related CPU;
– safety-related I/O card(s);
– non-safety-related Input card(s);

– fieldbus (allowing functional safety-related communication according to IEC 61784-3 (all parts));
– safety-related converter (according to IEC 61800-5-2).

Those components represent pre-designed subsystems provided by a component manufacturer(s).

The converters provide the integrated safety-related sub-functions STO (Safe Torque Off) and SLS (Safely Limited Speed) according to IEC 61800-5-2.

Table F.4 shows the relevant signals to perform the safety functions which should be controlled and tested depending on the hardware wiring and the software implementation.

Table F.4 – Relevant list of input and output signals

List of input signals				
Description (function, signal)	**Variable (designation)**	**Address (designation)**	**HW wiring correct (y/n)**	**SW interconnection correct (y/n)**
GD1, contact 1 (NC)	IS_bGD1_1	I8.0		
GD1, contact 2 (NO)	IS_bGD1_2	I9.4		
GD2, contact 1 (NC)	IS_bGD2_1	I8.1		
GD2, contact 2 (NO)	IS_bGD2_2	I9.5		
ES1, two contacts (NC)	IS_ES1	I8.4 (I10.0)		
ES2, two contacts (NC)	IS_ES2	I8.5 (I10.1)		
M3, feedback contactors (NC)	IS_bSTAT_M3	I8.6		
EN1, enabling contact 1 (NO)	IS_bEN1_1	I9.0 (I10.4)		
EN1, enabling contact 2 (NO)	IS_bEN1_2	I9.0 (I10.4)		
EN2, enabling contact 1 (NO)	IS_bEN2_1	I9.0 (I10.4)		
EN2, enabling contact 2 (NO)	IS_bEN2_2	I9.0 (I10.4)		
ACK1, acknowledge contact (NO)	I_bACK1	I4.0		
ACK2, acknowledge contact (NO)	I_bACK2	I5.0		
List output signals				
M1, STO	QS_bM1_STO	Q32.0		
M1, SLS	QS_b12_SLS	Q32.4		
M2, STO	QS_bM2_STO	Q48.0		
M2, SLS	QS_bM2_SLS	Q48.4		
M3, two contactors	QS_bM3	Q24.0		
Date:				
Name:				
Software signature:				
Hardware signature:				
NOTE 1 NC means normally closed and NO normally open. NOTE 2 The controlling of the hardware and software design can be executed by different persons. The confirmation by one person that those controlling activities were executed is important.				

F.5 Software system design specification

Figure F.2 shows the software system design based on principal design of the module architecture.

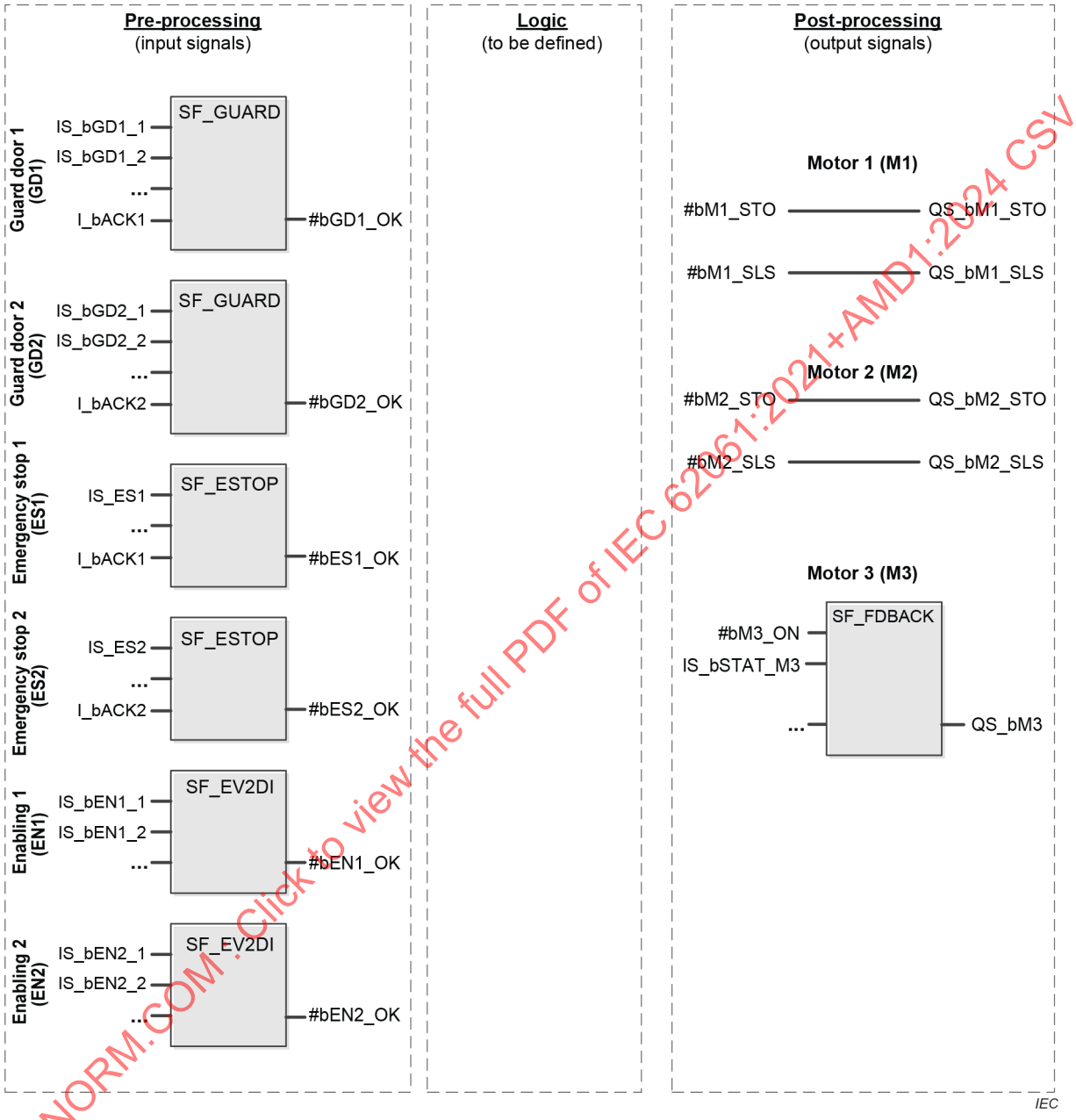


Figure F.2 – Principal module architecture design

The following pre-designed function blocks (library) are used:

– *SF_GUARD:* Monitoring of a guard door by two interlocking devices (e.g. position switches with NC contacts) *IS_bGDx_1* and *IS_bGDx_2* (discrepancy time control realised by the function block); when the state of one of these two input signals is equal to 0, then *#bGDx_OK* is set to 0; *#bGDx_OK* will be set to 1 by closing the guard door and after this applying a rising edge at *I_ACKx*.

– *SF_ESTOP:* Monitoring of two NC contacts of the emergency stop device *IS_ES1* and *IS_ES2* (discrepancy time control realised by the input card); when the state of one of these two input signals is equal to 0, then *#bESx_OK* is set to 0; *#bESx_OK* will be set to 1 by unlatching the emergency stop device and after this applying a rising edge at *I_ACKx*.

– *SF_EV2DI:* Monitoring of two NC contacts of the enabling device *IS_bENx_1* and *IS_bENx_2* (discrepancy time control realised by the function block); when the state of one of these two input signals is equal to 0, then *#bENx_OK* is set to 0; *#bENx_OK* will be set to 1 automatically by releasing the enabling device.

– *SF_FDBACK:* Monitoring of contactors by using the mirror contacts (as feedback); a feedback error is detected if the inverse signal state of the feedback input *IS_STAT_M3* does not follow the signal state of output *QS_M3* within the maximum tolerable feedback time.

NOTE The reset of any failures, e.g. discrepancy time or feedback error is not shown here.

Figure F.3 shows the principal design approach of the logic layer.

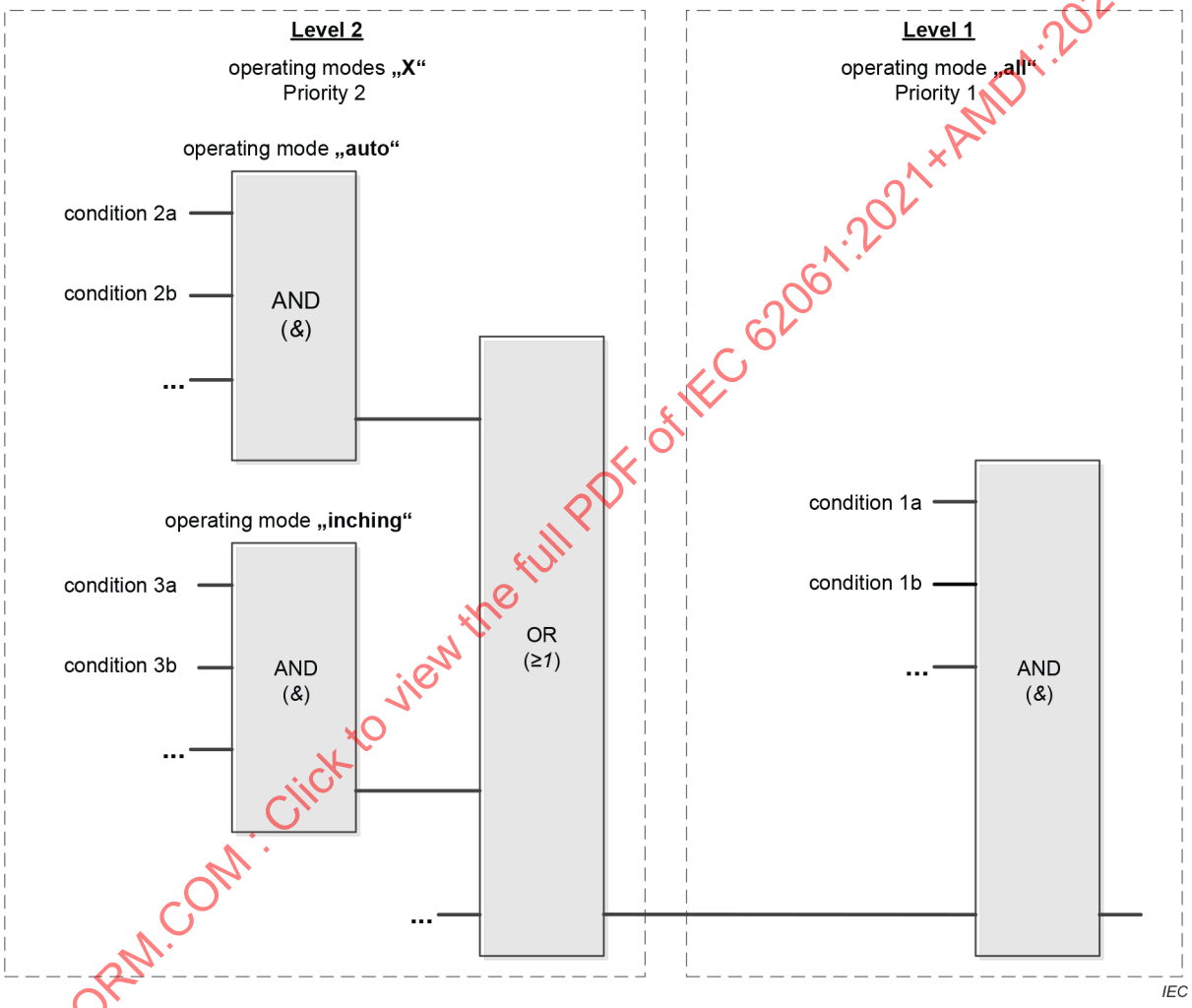
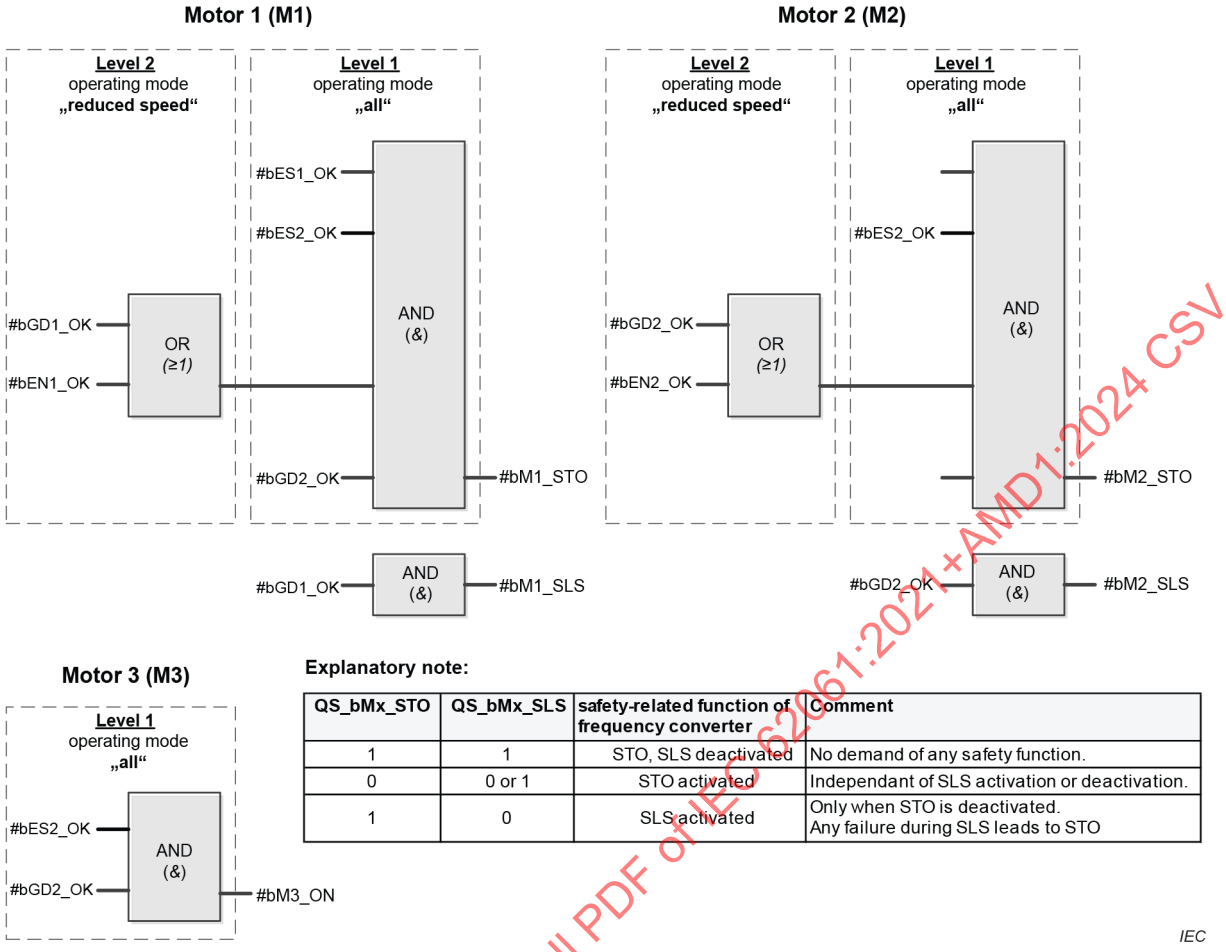


Figure F.3 – Principal design approach of logical evaluation

Figure F.4 represents logic evaluation of the safety functions described in Table F.3.



Explanatory note:

QS_bMx_STO	QS_bMx_SLS	safety-related function of frequency converter	Comment
1	1	STO, SLS deactivated	No demand of any safety function.
0	0 or 1	STO activated	Independant of SLS activation or deactivation.
1	0	SLS activated	Only when STO is deactivated. Any failure during SLS leads to STO

Figure F.4 – Example of logical representation (program sketch)

Alternatively, a simplified cause and effect matrix can be used, showing all the safety functions with the corresponding input(s) which will initiate the safety functions (causes or initiation events) and switched output(s) (effects), see Table F.5. There are different types of representation of a cause and effect matrix. The more convenient can be used.

Table F.5 – Example of simplified cause and effect matrix

Safety functions	Inputs	M1_STO	M1_SLS	M2_STO	M2_SLS	M3_ON
SF1	ES1_OK	&				
SF2	ES2_OK	&		&		&
SF10	GD1_OK		&			
SF11	GD2_OK	&			&	&
SF20	GD1_OK or EN1_OK	&				
SF21	GD2_OK or EN2_OK			&		
	Causes	**Effects**				

F.6 Protocols

Table F.6 and Table F.7 show the protocol of verification of the software system design specification and protocol of software code review. It represents only an important summary of already executed verifications.

Table F.6 – Verification of software system design specification

to be checked	reference	correct (y/n)
1. Does the module architecture comply with the specification of the safety functions?	Figure F.3	
2. Does the software design specification comply with the specification of the safety functions?	Table F.3 Figure F.3 Figure F.4	
Date:		
Name:		
Software signature:		
Hardware signature:		

Table F.7 – Software code review

to be checked	reference	correct (y/n)
1. Does the software comply with the coding guidelines?	Table F.2	
2. Does the control system design comply with the specification?		
3. Is the interconnection of the I/O-signals in the software correct? Is the parameterization of the relevant FBs correct?		
4. Does the hierarchy of the PLC-safety-program comply with the specification?		
5. Does the architecture of PLC-safety-program comply with the specification?		
6. Does the PLC-safety-program comply with the table specification?		
7. Does the safety-related software specification comply with the specification of the safety functions?		
Date:		
Name:		
Software signature:		
Hardware signature:		

The software validation (see Table F.8) is partially a summary of already executed tests. Additional manufacturer specific tests of e.g. the correct parameterization of external safety devices like laser scanners, converts, light curtains etc. are required. In the example under consideration, the threshold of the safely limited speed of the converter (and other parameters) has to be checked. These manufacturer specific tests are not shown here. The required documentation listed in Table F.8 can be archived electronically. This documentation is important e.g. for an external certification of the machine.

Table F.8 – Software validation

to be checked	reference	correct (y/n)
1. Was the I/O-test carried out with a positive result?	Table F.4	
2. Was the test of the safety functions and other test requirements carried out with a positive result?	Table F.4 Figure F.3 Figure F.4	
3. Were all manufacturer specific tests of the parameterization of external safety devices (e.g. laser scanners, converters …) carried out positively and documented?		
necessary documentation needed	**reference**	**existent (y/n)**
4. Documents of the V-model		
5. Final document of the safety relevant software including signatures		
6. Final document of the control system hardware configuration with checksums and all adjustments		
7. Archiving of the handbooks of all safety relevant system components		
8. Final document of the configuration of all safety relevant peripheral devices		
9. The relevant C standards		
Date:		
Name:		
Software signature:		
Hardware signature:		

Annex G
(informative)

Examples of safety functions

Examples of typical safety functions with some references to international standards are given in Table G.1.

NOTE The list of safety functions and standards in Table G.1 is not exhaustive.

Table G.1 – Examples of typical safety functions

Safety function	ISO 12100: 2010	Standards and information
Safety-related stopping, initiated by a – guard – protective device	6.2.11.3	IEC 60204-1, stop categories IEC 61800-5-2, drive functions, e.g. STO, SS1, SBC ISO 14119, Interlocking devices associated with guards IEC 61496 (all parts), Electro-sensitive protective equipment
Start and re-start (see Note 1)	6.2.11.3	IEC 60204-1, ISO 14118
Manually operated control system (manual handling)	6.2.11.8	IEC 60204-1, Type C standards ISO 11161 Integrated manufacturing systems Type C standards
– device with reset (push button)	6.2.11.8 b)	
– two-hand control	6.2.11.8 b) 6.2.11.9	ISO 13851 Two-hand control devices
– hold-to-run devices	6.2.11.8 b)	
Adjusting, teaching, retooling, fault finding, maintenance, cleaning		IEC 60204-1, Type C standards
– enabling function		IEC 60947-5-8
– safe motion, safe “positioning”		IEC 61800-5-2, drive functions, e.g. SLS, SOS
Selection of control or operating modes (see Note 2)	6.2.11.10	9.2.3.5 of IEC 60204-1:2016; 8.4 of ISO 11161:2007, (Integrated manufacturing systems); Type C standards
Guard locking	3.27.5	ISO 14119 Interlocking devices associated with guards
Emergency stop (see Note 3)		IEC 60204-1, stop categories; ISO 13850, emergency stop functions; IEC 60947-5-5, mechanical latching function
NOTE 1 To be considered in interrelationship to “unexpected starting”. NOTE 2 In general to be evaluated in interrelationship to machine functions (e.g. selection of routine in the application software of the safety controller) and requirements of systematic integrity. NOTE 3 Complementary protective measures refer to ISO 12100.		

Annex H
(informative)

Simplified approaches to evaluate the *PFH* value of a subsystem

H.1 Table allocation approach

The following procedure allows evaluating the PFH value of a subsystem:

1) Selection of the used architecture of a not-pre-designed subsystem based on the *DC*(s) per channel;

   NOTE 1 A pre-designed subsystem is characterized by a SIL with a PFH value (see also 6.2). A not-pre-designed subsystem claims a maximum SIL based on the architectural constraints (see 7.4).

   Where the *DC*s per channel are different, either the lowest *DC* per channel may be used as a worst case approach, or the arithmetic average of *DC* per channel of both channels.

2) Determination of PFH value with Table H.1 and Table H.2 for not-pre-designed subsystems:
   - using Table H.1 for components qualified by $MTTF_D$ per channel and DC to allocate the PFH value within a range of 10 %, 20 %, 30 %, 40 % or 50 % of the limit of the respective required SIL, or
   - using Table H.2 for components qualified by B_{10D} and equation (7) in 7.3.4.2 to determine the $MTTF_D$ per channel and then, by use of Table H.1, allocating the PFH value within a range of 10 %, 20 %, 30 %, 40 % or 50 % of the limit of the respective required SIL.

NOTE 2 Where a dual channel architecture is used and the $MTTF_D$ per channel are different, either the lowest $MTTF_D$ per channel of both channels can be used as a worst case approach, or the geometric average of $MTTF_D$ per channel of both channels. Example: $MTTF_{De1}$ = 20 a, $MTTF_{De2}$ = 200 a. The geometric average is calculated as follows: $MTTF_D = \sqrt{20\text{ a} \times 200\text{ a}} = 63{,}2\text{ a}$.

The following numerical examples show the use of the table allocation approach:

- with DC_1 = 90 %, DC_2 = 90 % and $MTTF_D$ = 60 years per channel, 0,1 × 10^{-06} = 1 × 10^{-07} as 10 % of the PFH value for SIL 2 can be allocated;
- with DC_1 = 90 %, DC_2 ≥ 99 % and $MTTF_D$ = 50 years per channel, 0,2 × 10^{-06} = 2 × 10^{-07} as 20 % of the PFH value for SIL 2 with DC ≥ 90 % can be allocated;
- with DC_1 = 90 %, DC_2 = 90 % and $MTTF_D$ = 20 years per channel, 0,3 × 10^{-05} = 3 × 10^{-06} as 30 % of the PFH value for SIL 1 with DC ≥ 60 % can be allocated;
- with DC_1 = 99 %, DC_2 = 99 %, $MTTF_{D1}$ = 20 years and $MTTF_{D2}$ = 200 years, $MTTF_D$ = 63,2 years per channel can be used and 0,5 x 10^{-07} = 5 × 10^{-08} as ~~30~~ 50 % of the PFH value for SIL 3 can be allocated.

Table H.1 – Allocation of *PFH* value of a subsystem

$MTTF_D$ per channel and DC														
Single channel architecture				Dual channel architecture										
$MTTF_D$ [years]	DC	$MTTF_D$ [years]	DC	$MTTF_D$ [years]	DC	$MTTF_D$ [years]	DC	$MTTF_D$ [years]	DC		PFH [h^{-1}]			
23 – < 29	0 %	17 – < 20	60 %	21 – < 24	0 %	13 – < 15	60 %			→	50 %	5×10^{-6}	SIL 1	10^{-5}
29 – < 38	0 %	20 – < 25	60 %	24 – < 27	0 %	15 – < 17	60 %			→	40 %	4×10^{-6}		
38 – < 57	0 %	25 – < 33	60 %	27 – < 34	0 %	17 – < 22	60 %			→	30 %	3×10^{-6}		
57 – < 114	0 %	33 – < 58	60 %	34 – < 48	0 %	22 – < 31	60 %			→	20 %	2×10^{-6}		
≥ 114	0 %	≥ 58	60 %	≥ 48	0 %	≥ 31	60 %			→	10 %	1×10^{-6}		10^{-6}
		60 – < 69	90 %			23 – < 26	90 %	9 – < 11	99 %	→	50 %	5×10^{-7}	SIL 2	
		69 – < 84	90 %			26 – < 31	90 %	11 – < 13	99 %	→	40 %	4×10^{-7}		
		84 – < 112	90 %			31 – < 39	90 %	13 – < 18	99 %	→	30 %	3×10^{-7}		
		112 – < 187	90 %			39 – < 60	90 %	18 – < 30	99 %	→	20 %	2×10^{-7}		
		≥ 187	90 %			≥ 60	90 %	≥ 30	99 %	→	10 %	1×10^{-7}		10^{-7}
								54 – < 65	99 %	→	50 %	5×10^{-8}	SIL 3	
								65 – < 85	99 %	→	40 %	4×10^{-8}		
								85 – < 123	99 %	→	30 %	3×10^{-8}		
								123 – < 238	99 %	→	20 %	2×10^{-8}		
								≥ 238	99 %	→	10 %	1×10^{-8}		10^{-8}
A		C		B		D		D						
Basic subsystem architecture														

NOTE 3 Table H.1 is based on the formulas described in Clause H.2. A common cause failure factor of β = 2 % and a useful lifetime of 20 years have been assumed.

NOTE 4 In case of architecture C, the diagnostic channel was assumed to have the same $MTTF_D$ as the functional channel. The diagnostic channel can have a $MTTF_D$ down to the half of the $MTTF_D$ of the functional channel. The consequent increase of the actual PFH from the table value remains below an acceptable limit. Moreover, time-optimal monitoring is assumed, see NOTE in H.2.4.1.

NOTE 5 In case of architecture D, the diagnostic test interval T_2 was assumed to not exceed 1 week.

NOTE 6 In case of architecture C, a fault handling function is assumed. Its realisation can have at least the half $MTTF_D$ of the functional channel.

NOTE 7 One can claim a lower DC than actually achieved (see 3rd numerical example above).

Table H.2 – Relationship between B_{10D}, operations and $MTTF_D$

interval or operations		
cycle interval	duty cycle (per hour)	annual operations n_{op}
30 sec	120	1.051.200
1 min	60	525.600
2 min	30	262.800
5 min	12	105.120
10 min	6	52.560
15 min	4	35.040
30 min	2	17.520
1 h	1	8.760
2 h	0,5	4.380
>= 4 h	0,25	2.190

$MTTF_D$ per channel [years] based on operations and B_{10D}								
B_{10D}								
100.000	400.000	1.000.000	1.300.000	2.000.000	5.000.000	10.000.000	20.000.000	50.000.000
not recommended		10	12	19	48	95	190	476
		19	25	38	95	190	381	951
	15	38	49	76	190	381	761	1903
10	38	95	124	190	476	951	1903	
19	76	190	247	381	951	1903		
29	114	285	371	571	1427			
57	228	571	742	1142				
114	457	1142	1484					
228	913	2283						
457	1826							

$MTTF_D$ > 2000 years

interval or operations based on:
24 hours per day
365 days per year
(1 year ≈ 8760 h)

$T10_D \approx MTTF_D / 10$

sensors	emergency stop devices	emergency stop devices	position switch (with separate actuator, guard-locking)		position switches (with separate actuator, guard-locking)	position switches	position switches	position switches	position switches
	enabeling switches						pushbuttons	pushbuttons	pushbuttons
		proximity switches with nominal load						proximity switches with small load	
actuators	circuit breakers	relays and contactor relays with nominal load		contactors with nominal load				relays and contactor relays with small load	
								contactors with small load	
								pneumatic components	

H.2 Simplified formulas for the estimation of PFH

H.2.1 General

This clause describes a simplified approach to the estimation of PFH for a number of basic subsystem architectures and gives formulas that can be used for subsystems. The formulas are in themselves a simplification and are intended to provide estimates that are biased towards the safe direction. The precondition for the validity for all formulas given in this clause is that the subsystem is operating in the "high demand or continuous mode".

NOTE For equations given in H.2.2 to H.2.5, constant and sufficiently low ($1 >> \lambda \times T_1$) failure rates (λ) of the subsystem elements are assumed (this means that the mean time to dangerous failure is much greater than the proof test interval or the useful lifetime of the subsystem).

H.2.2 Basic subsystem architecture A: single channel without a diagnostic function

This single channel subsystem covers the architecture A subsystem of 7.5.2.1. In this architecture (see Figure H.1), any dangerous failure of a subsystem element causes a failure of the safety function.

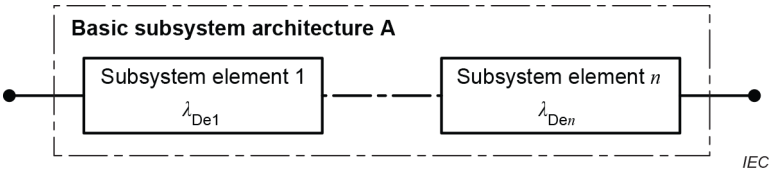


Figure H.1 – Basic subsystem architecture A logical representation

For architecture A, the PFH of the subsystem is the sum of the dangerous failure rates of all subsystems elements:

$$PFH = \lambda_{De1} + \cdots + \lambda_{Den} \quad \text{(H.1)}$$

where

λ_{Dei} is the dangerous failure rate of element ei within the single functional channel.

H.2.3 Basic subsystem architecture B: dual channel without a diagnostic function

This dual channel subsystem covers the architecture B subsystem of 7.5.2.2. This architecture (see Figure H.2) is such that a single failure of any subsystem element does not cause a loss of the safety function. Thus, there would have to be a dangerous failure in more than one element before failure of the safety function can occur.

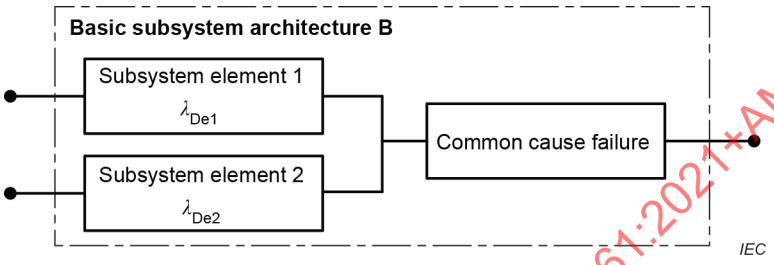


Figure H.2 – Basic subsystem architecture B logical representation

For architecture B, the PFH of the subsystem is:

$$PFH = (1-\beta)^2 \times \lambda_{De1} \times \lambda_{De2} \times T_1 + \beta \times (\lambda_{De1} + \lambda_{De2}) / 2 \quad \text{(H.2)}$$

where

λ_{De1} is the dangerous failure rate of element e1 comprising the first functional channel;

λ_{De2} is the dangerous failure rate of element e2 comprising the second functional channel;

T_1 is the proof test interval of the perfect proof test or useful lifetime, whichever is the smaller;

β is the susceptibility to common cause failures factor.

H.2.4 Basic subsystem architecture C: single channel with a diagnostic function

H.2.4.1 General

This single channel subsystem covers the architecture C subsystem of 7.5.2.3 (see Figure H.3).

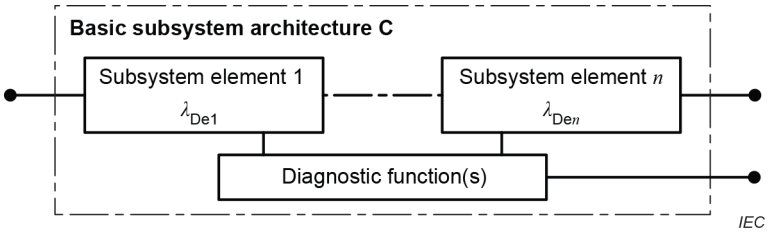


Figure H.3 – Basic subsystem architecture C logical representation

The safety function is performed by a single channel comprising the elements e1 to en. Any undetected dangerous fault of a subsystem element leads to a dangerous failure of the safety function.

Where a fault of a subsystem element is detected, the diagnostic function(s) initiates a fault reaction function (see ~~7.4.3~~ 7.4.3.2).

In the following, the notion of fault handling function is used. The fault handling function comprises both the fault detection function and the fault reaction function, see Figure H.4.

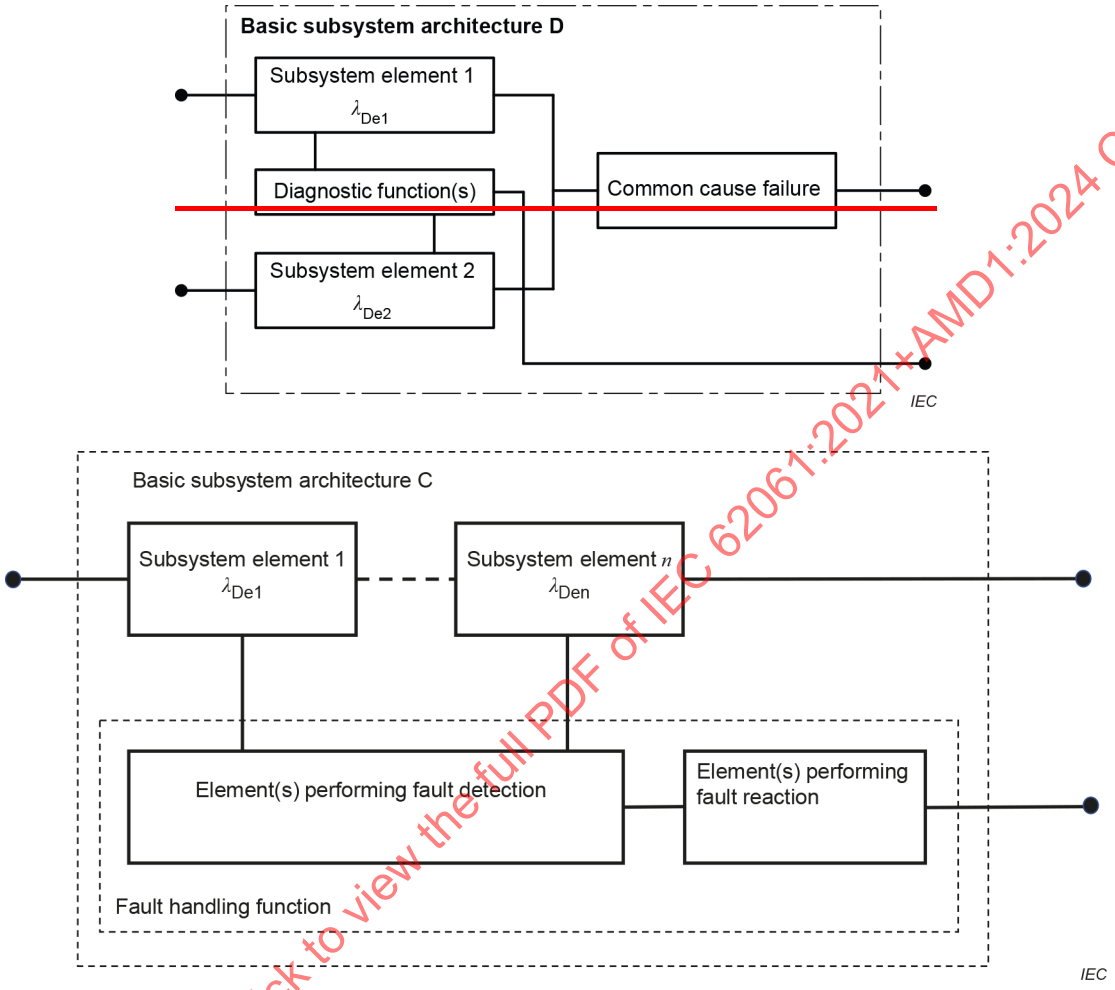


Figure H.4 – Correlation of basic subsystem architecture C and the pertinent fault handling function

All approaches of H.2.4 for the calculation of PFH assume time-optimal fault handling. Time-optimal fault handling of a subsystem element can be assumed if:

- the diagnostic rate is at least a factor of 100 higher than the demand rate of the safety function and the time needed for the fault reaction is sufficiently short to bring the system to a safe state before a hazardous event occurs; or
- the fault handling is performed immediately upon any potential demand of the safety function and the time needed to detect a detectable fault and to bring the system to a safe state is shorter than the process safety time; or
- the fault handling is performed continuously and the time needed to detect a detectable fault and to bring the system to a safe state is shorter than the process safety time; or
- the fault handling is performed periodically and the sum of the test interval, the time needed to detect a detectable fault and time needed to bring the system to a safe state is shorter than the process safety time.

NOTE Although the failure of the fault handling function will not cause a failure of the safety function, the elements contributing to the fault handling function are assigned a dangerous failure rate containing the letter D in the index of λ. Dangerous failures in this sense are failures that lead to a loss of the fault handling function. The dangerous

failure rate of elements involved in the fault handling function does not cover failures which lead to a fault reaction although there is no failure of the functional channel (so-called “false trips”).

H.2.4.2 External fault handling function

The fault handling function may be completely performed by a separate subsystem(s) of the SCS which is also involved in performing the safety function, thus contributing to its PFH. These conditions are depicted in Figure H.5.

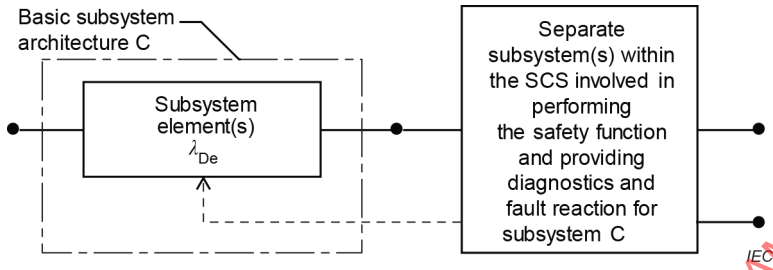


Figure H.5 – Basic subsystem architecture C with external fault handling function

~~Subsystem C may be comprised of n elements numbered from 1 to n.~~

If the diagnostic function and the fault reaction function are provided by separate subsystem(s) within the SCS, the PFH of the subsystem is:

$$PFH = (1 - DC_1) \times \lambda_{De1} + \cdots + (1 - DC_n) \times \lambda_{Den} \quad \text{(H.3)}$$

where

λ_{De1} is the dangerous failure rate of the first element e1 within the single functional channel;

λ_{Den} is the dangerous failure rate of the nth element en within the single functional channel;

DC_1 is the diagnostic coverage for the first element e1 of the single functional channel;

DC_n is the diagnostic coverage for the nth element en of the single functional channel;

n is the number of elements of the single functional channel.

NOTE PFH estimation shown above only related to subsystem C. For the complete safety function, other factors are taken into account, see 6.4.

H.2.4.3 Fault handling partially or completely done within the subsystem

For the following case, the notion of a fault handling function is needed with the dangerous failure rate $\lambda_{D\ FH}$ associated to it. It is defined as follows:

“In case of fault handling partially or completely done within the subsystem, the notion of a fault handling function is needed with the dangerous failure rate $\lambda_{D\ FH}$ associated to it. There are the following three cases:”

- If the diagnostic function is provided by a separate subsystem within the SCS and the fault reaction function is provided by this architecture C subsystem, the reaction function is comprised by the fault reaction function only and $\lambda_{D\ FH} = \lambda_{D\ FR}$. These conditions are depicted in Figure H.6.
- If the diagnostic function is provided by this architecture C subsystem and the fault reaction function is provided by a separate subsystem within the SCS, the reaction function is comprised by the diagnostics reaction function only and $\lambda_{D\ FH} = \lambda_{D\ FD}$. These conditions are depicted in Figure H.7.

- If the diagnostic function and the fault reaction function are both provided by this architecture C subsystem, the reaction function is comprised by the diagnostic function and the fault reaction function and $\lambda_{D\ FH} = \lambda_{D\ FD} + \lambda_{D\ FR}$. These conditions are depicted in Figure H.8.

In all three cases, the dangerous failure rate of the fault handling function $\lambda_{D\ FH}$ is given by the sum of the dangerous failure rates of all elements which contribute to the fault handling function and which are part of subsystem C.

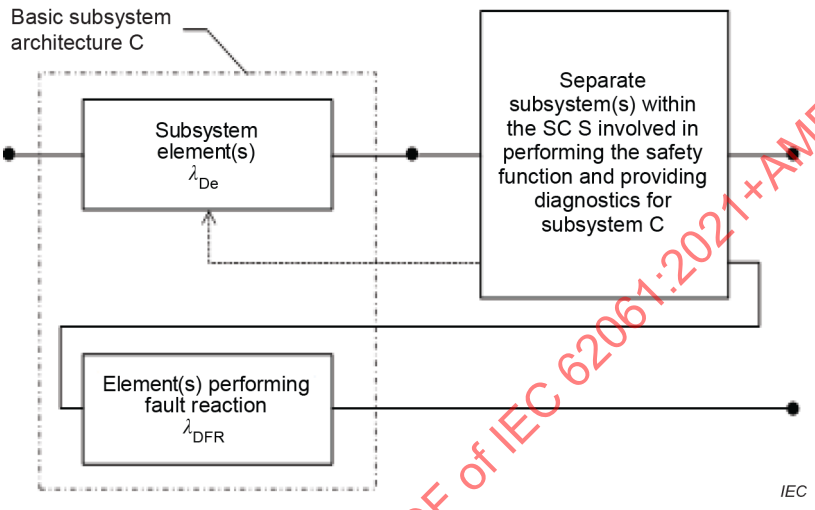


Figure H.6 – Basic subsystem architecture C with external fault diagnostics

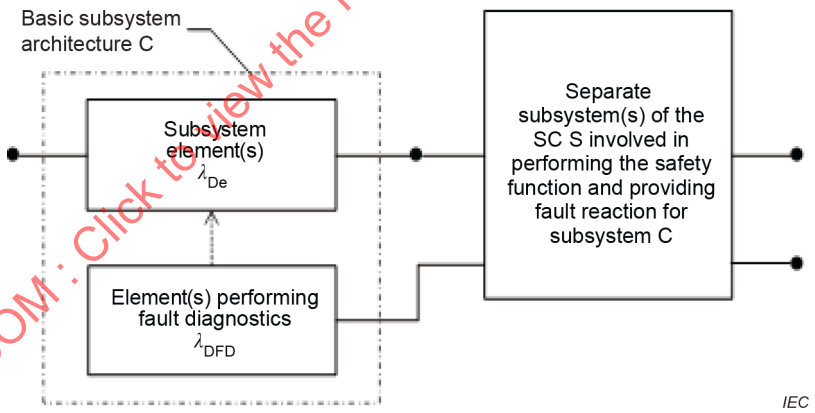


Figure H.7 – Basic subsystem architecture C with external fault reaction

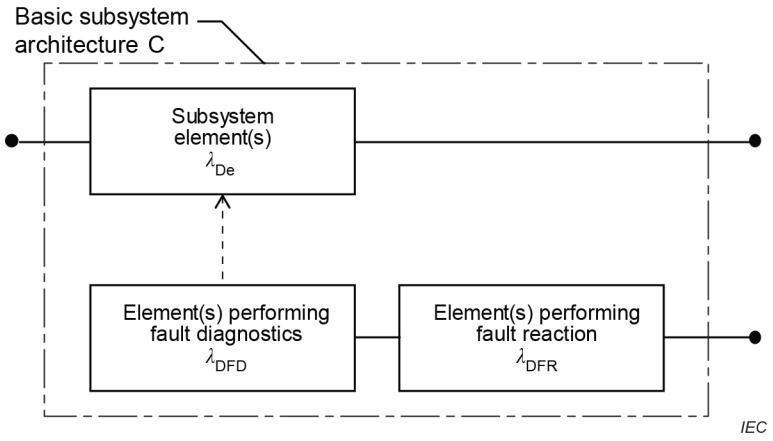


Figure H.8 – Basic subsystem architecture C with internal fault diagnostics and internal fault reaction

If in one of the three above-described cases, all of the following conditions apply

- $\beta \leq 2$ %;
- $DC \leq 99$ %;
- $1/\lambda_{De} \leq 1\,000$ years;
- $1/\lambda_{D\,FH}$ has at least the minimum value according to Table H.3;

where $\lambda_{D\,FH}$ is the failure rate of the single element that realizes the fault handling function(s) within the subsystem, then the PFH value of the subsystem C can be calculated using equation (H.43).

NOTE 1 Common cause failures factor (β) is considered between the channel (λ_{De}) and the fault ~~reaction~~ handling function channel (~~λ_{DFR}~~ $\lambda_{D\,FH}$).

Table H.3 – Minimum value of $1/\lambda_{D\,FH}$ for the applicability of PFH equation (H.43)

DC range	Minimum value of $1/\lambda_{D\,FH}$ (years)
60 % ≤ DC < 65 %	44
65 % ≤ DC < 70 %	59
70 % ≤ DC < 75 %	100
75 % ≤ DC < 80 %	170
80 % ≤ DC < 85 %	300
85 % ≤ DC < 90 %	550
90 % ≤ DC < 95 %	1 200
95 % ≤ DC ≤ 99 %	5 900
NOTE If the functional channel is comprised by more than one element, the related DC is calculated by using Equation (H.6).	

If at least one of the above conditions is not fulfilled, then one of the following approaches can be used. These approaches are also applicable if the conditions are fulfilled.

If the functional channel is comprised by one element only and the fault handling function(s) within the subsystem is (are) realized by another single element, the following equation can be used to calculate PFH:

$$PFH = \lambda_{De} - DC \times \left[\lambda_{De} - \beta \times \min\left(\lambda_{De}, \lambda_{D\,FH}\right)\right] \times \left\{1 - \frac{1}{2}\left[\lambda_{D\,FH} - \beta \times \min\left(\lambda_{De}, \lambda_{D\,FH}\right)\right] \times T_1\right\} \quad \text{(H.4)}$$

where

T_1 is the proof test interval of the perfect proof test or useful lifetime, whichever is the smaller;

λ_{De} is the dangerous failure rate of the single element e of the functional channel;

$\lambda_{D\,FH}$ is the failure rate of the single element that realizes the fault handling function(s) within the subsystem;

DC is the diagnostic coverage for the single element e of the functional channel;

β is the susceptibility to common cause failures of the functional channel and the channel that realizes the fault handling function(s) within the subsystem.

If the functional channel is comprised by n elements and the fault handling function(s) within the subsystem is (are) realized by m elements, the following equations can be used to calculate PFH:

$$PFH = \sum_{i=1}^{n} \lambda_{Dei} - DC \times \left(\sum_{i=1}^{n} \lambda_{Dei} - \lambda_{CC}\right) \times \left\{1 - \frac{1}{2}\left[\sum_{j=1}^{m} \lambda_{D\,FH\,j} - \lambda_{CC}\right] \times T_1\right\} \quad \text{(H.5)}$$

with

$$DC = \frac{\sum_{i=1}^{n} (DC_i \times \lambda_{Dei})}{\sum_{i=1}^{n} \lambda_{Dei}} \quad \text{(H.6)}$$

$$\lambda_{CC} = \beta \times \min\left(\sum_{i=1}^{n} \lambda_{Dei}, \sum_{j=1}^{m} \lambda_{D\,FH\,j}\right) \quad \text{(H.7)}$$

where

T_1 is the proof test interval of the perfect proof test or useful lifetime, whichever is the smaller;

λ_{Dei} is the dangerous failure rate of element ei within the single functional channel;

n is the number of elements of the single functional channel;

$\lambda_{D\,FH\,j}$ is the failures rate of the element number j within the single channel that realizes the fault handling function(s) for the functional channel within the subsystem;

m is the number of elements of the single channel that realizes the fault handling function(s) for the functional channel within the subsystem;

DC_i is the diagnostic coverage for element ei of the single functional channel;

β is the susceptibility to common cause failures of the functional channel and the channel that realizes the fault handling function(s) for the functional channel within the subsystem.

NOTE 2 In case that the dangerous failure rate(s) of the fault handling function(s) within the subsystem can be assumed to be zero ($\lambda_{D\ FH\ j} = 0$), equations (H.5) to (H.7) simplify to equation (H.3).

NOTE 3 PFH estimation shown above only related to subsystem C. For the complete safety function, other factors are taken into account, see 6.4.

H.2.5 Basic subsystem architecture D: dual channel with a diagnostic function(s)

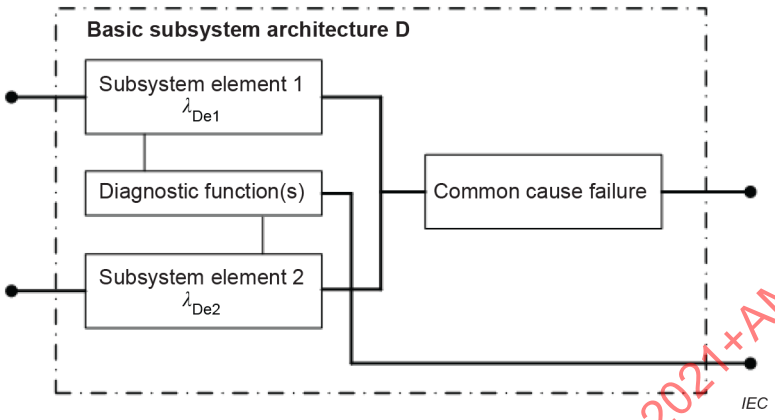


Figure H.9 – Basic subsystem architecture D logical representation

This dual channel subsystem covers the architecture D subsystem of 7.5.2.4. This architecture (see Figure H.9) is such that a single failure of any subsystem element does not cause a loss of the safety function.

For subsystem elements of different design, the PFH of the subsystem is:

$$PFH = (1-\beta)^2 \times \left[\lambda_{De1} \times \lambda_{De2} \times (DC_1 + DC_2) \times T_2 / 2 + \lambda_{De1} \times \lambda_{De2} \times (2 - DC_1 - DC_2) \times T_1 / 2\right] + \beta \times (\lambda_{De1} + \lambda_{De2}) / 2 \quad \text{(H.8)}$$

where

T_2 is the diagnostic test interval;

T_1 is the proof test interval of the perfect proof test or useful lifetime, whichever is the smaller;

β is the susceptibility to common cause failures;

λ_{De1} is the dangerous failure rate of subsystem element e1;

λ_{De2} is the dangerous failure rate of subsystem element e2;

DC_1 is the diagnostic coverage for subsystem element e1;

DC_2 is the diagnostic coverage for subsystem element e2.

For architecture D for subsystem elements of the same design, the PFH of the subsystem is:

$$PFH = (1-\beta)^2 \times \left[DC \times T_2 + (1-DC) \times T_1\right] \times \lambda_{De}{}^2 + \beta \times \lambda_{De} \quad \text{(H.9)}$$

where

λ_{De} is the dangerous failure rate of subsystem element e1 or e2;

DC is the diagnostic coverage for subsystem element e1 or e2.

H.3 Parts count method

Use of the “parts count method” serves to estimate the λ_D for each channel separately.

NOTE The parts count method is an approximation which always errs on the safe side. For more exact values, failure modes are required, but this can be very complicated.

$$\lambda_D = \sum_{i=1}^{N} \lambda_{Di} = \sum_{j=1}^{\tilde{N}} \left(n_j \lambda_{Dj} \right) \quad \text{(H.10)}$$

where

λ_D is the dangerous failure rate of the complete channel;

λ_{Di} is the dangerous failure rate of each component which has a contribution to the safety function;

N is the total number of components;

n_j is the number of components within a set of equal components;

λ_{Dj} is the dangerous failure rate of each component of set number j of equal components which have a contribution to the safety function;

$\tilde{N}$ is the number of sets of equal components.

Annex I
(informative)

The functional safety plan and design activities

I.1 General

This annex illustrates the relationship between activities, documentation and roles of involved personnel during the lifetime of a machine.

I.2 Example of a machine design plan including a safety plan

Figure I.1 illustrates a non-exhaustive example of a machine design plan including a safety plan.

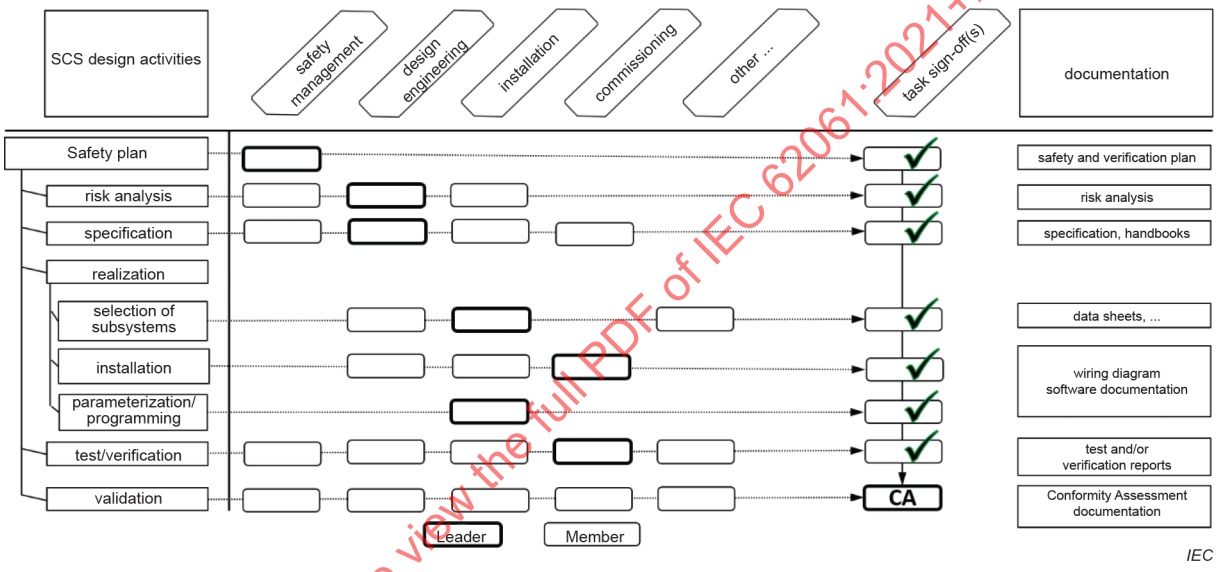
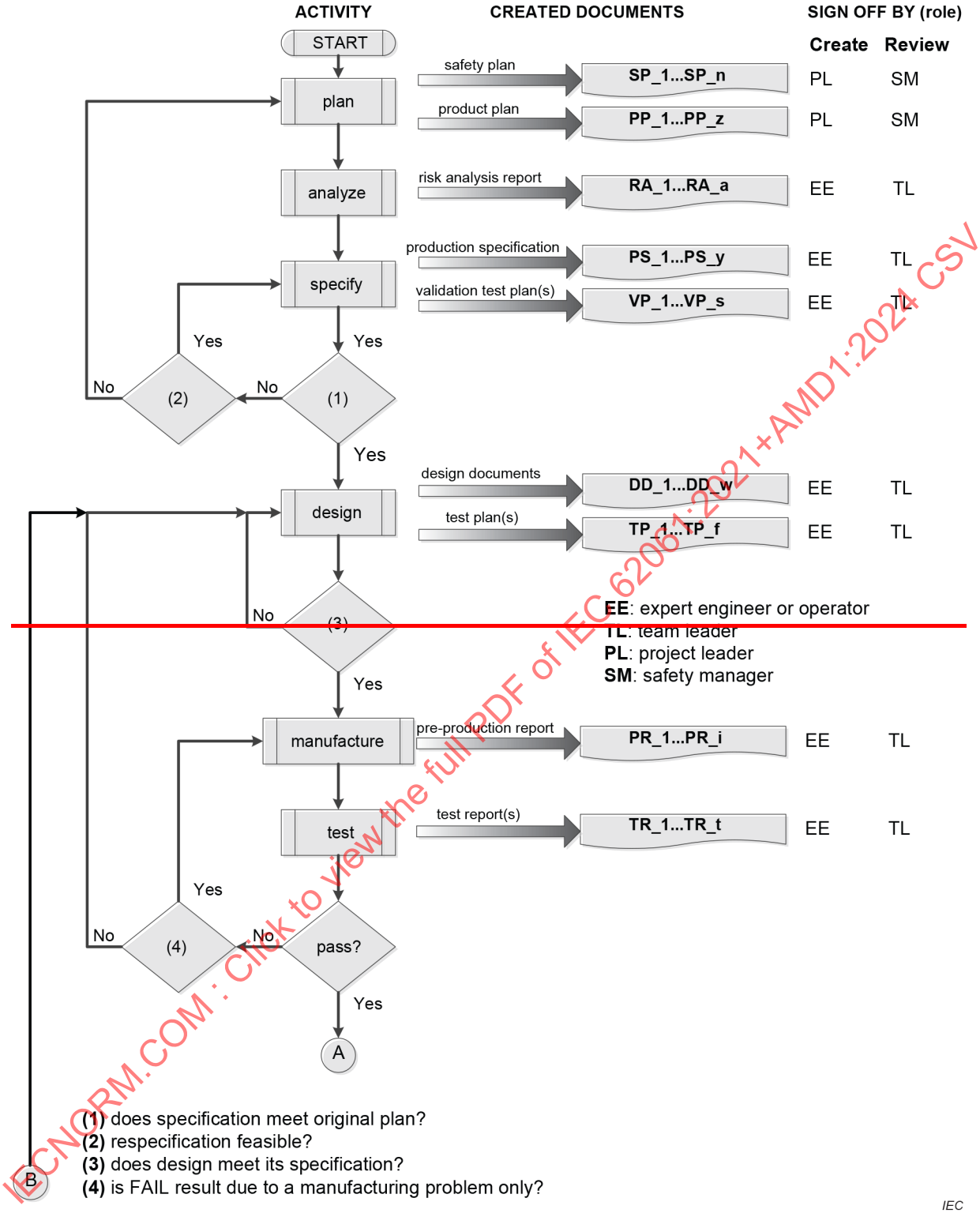


Figure I.1 – Example of a machine design plan including a safety plan

I.3 Example of activities, documents and roles

Figure I.2 illustrates example of activities, documentation and roles over the lifecycle.



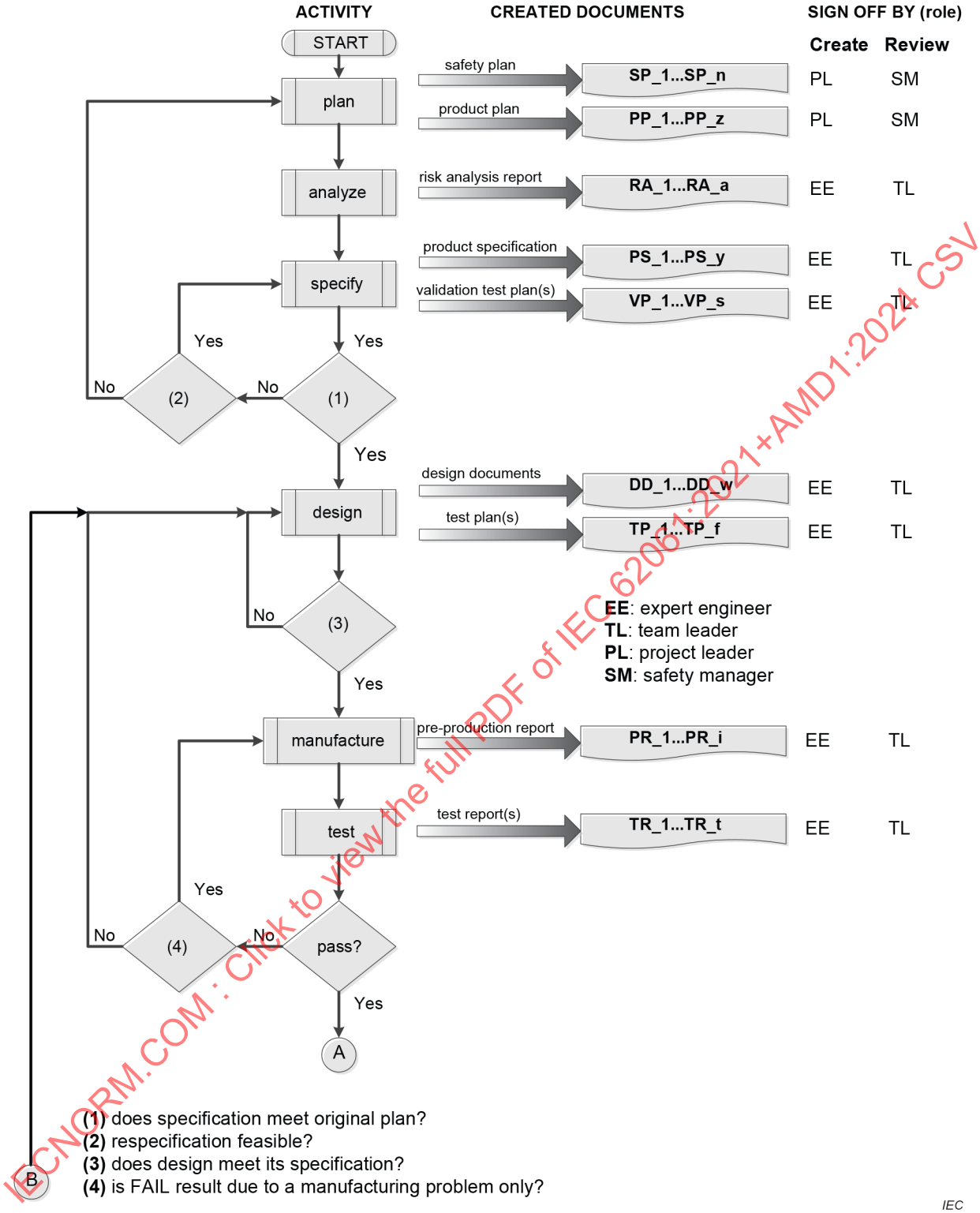


Figure I.2 – Example of activities, documents and roles *(1 of 2)*

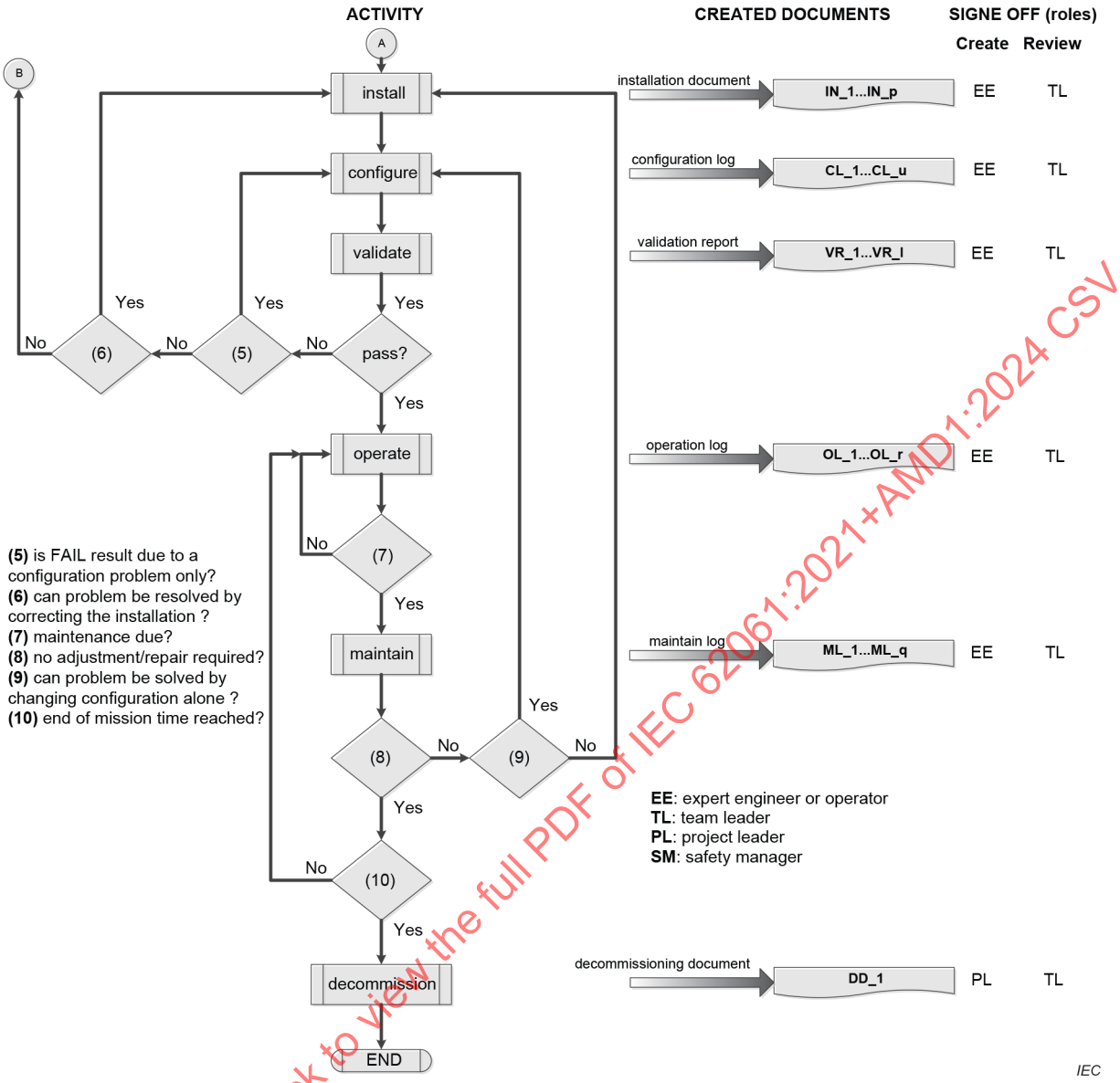


Figure I.2 *(2 of 2)*

Annex J
(informative)

Independence for reviews and testing/verification/validation activities

J.1 Software design

Where this document requires carrying out review or testing/verification activities, these should be performed by persons not involved directly in the design of the safety-related software, i.e. who are independent of the design process. The parties concerned may be other persons, departments or bodies who/which are not subordinate to the design department within the organization's hierarchy. The level of independence should be commensurate with the risk, i.e. with the required SIL, see Table J.1.

The level of independence specified in the table below is the minimum for the safety integrity level. Depending on a number of factors specific to the application (like previous experience, degree of complexity etc.), it could be appropriate to choose a higher level of independence.

Table J.1 – Minimum levels of independence for review, testing and verification activities

Minimum level of independence for review, testing and verification activities	SIL 1	SIL 2	SIL 3
Same person	not sufficient	not sufficient	not sufficient
Other person	not sufficient (see NOTE 2)	not sufficient (see NOTE 2)	not sufficient
Independent person	sufficient	sufficient	sufficient

An "independent person" may be involved in the same project, but should not be involved in the design activities and should not have responsibility for project management and should not have a superior role.

NOTE 1 Depending upon the company organisation and expertise within the company, the requirement for independent persons can have to be met by using an external organisation. Conversely, companies that have internal organisations skilled in risk assessment and the application of safety-related systems, that are independent of and separate (by ways of management and other resources) from those responsible for the main development, can be able to use their own resources to meet the requirements for an independent organisation.

NOTE 2 For software level 1 using combinations of pre-designed software modules only, an "other person" is sufficient.

NOTE 3 Software level 2 is not applicable for SIL 3, see 8.4.

J.2 Validation

The minimum level of independence of those carrying out validation should be as specified in Table J.2. The level of independence specified is the minimum for the safety integrity level.

Table J.2 – Minimum levels of independence for validation activities

Minimum level of independence for validation activities	SIL 1	SIL 2	SIL 3
Same person	not sufficient	not sufficient	not sufficient
Other person	not sufficient	not sufficient	not sufficient
Independent person	sufficient	sufficient	sufficient

An “independent person” may be involved in the same project, but should not be involved in the design activities and should not have responsibility for project management and should not have a superior role.

Bibliography

IEC 60050-192:2015, *International Electrotechnical Vocabulary (IEV) – Part 192: Dependability*

IEC 60068 (all parts), *Environmental testing*

IEC 60364-4-41:2005, *Low-voltage electrical installations – Part 4-41: Protection for safety – Protection against electric shock*
IEC 60364-4-41:2005/AMD1:2017

IEC 60449:1973[1], *Voltage bands for electrical installations of buildings*

IEC 60529, *Degrees of protection provided by enclosures (IP Code)*

IEC 60721 (all parts), *Classification of environmental conditions*

IEC 60812, *Failure modes and effects analysis (FMEA and FMECA)*

IEC 60947 (all parts), *Low-voltage switchgear and controlgear*

IEC 60947-4-1:2018, *Low-voltage switchgear and controlgear – Part 4-1: Contactors and motor-starters – Electromechanical contactors and motor-starters*

IEC 60947-5-1, *Low-voltage switchgear and controlgear – Part 5-1: Control circuit devices and switching elements – Electromechanical control circuit devices*

IEC 60947-5-3, *Low-voltage switchgear and controlgear – Part 5-3: Control circuit devices and switching elements – Requirements for proximity devices with defined behaviour under fault conditions (PDDB)*

IEC 60947-5-5, *Low-voltage switchgear and controlgear – Part 5-5: Control circuit devices and switching elements – Electrical emergency stop device with mechanical latching function*

IEC 60947-5-8, *Low-voltage switchgear and controlgear – Part 5-8: Control circuit devices and switching elements – Three-position enabling switches*

IEC 60950-1, *Information technology equipment – Safety – Part 1: general requirements*

IEC 61000-6-7, *Electromagnetic compatibility (EMC) – Part 6-7: Generic standards – Immunity requirements for equipment intended to perform functions in a safety-related system (functional safety) in industrial locations*

IEC 61025:2006, *Fault tree analysis (FTA)*

IEC 61131-2:2017, *Industrial-process measurement and control – Programmable controllers – Part 2: Equipment requirements and tests*

IEC 61131-6:2012, *Programmable controllers – Part 6: Functional safety*

IEC 61140:2016, *Protection against electric shock – Common aspects for installation and equipment*

[1] Withdrawn.

IEC 61165, *Application of Markov techniques*

IEC 61204-7:2016, *Low-voltage switch mode power supplies – Part 7: Safety requirements*

IEC 61310 (all parts), *Safety of machinery – Indication, marking and actuation*

IEC 61326-3-1, *Electrical equipment for measurement, control and laboratory use – EMC requirements – Part 3-1: Immunity requirements for safety-related systems and for equipment intended to perform safety-related functions (functional safety) – General industrial applications*

IEC 61496 (all parts), *Safety of machinery – Electro-sensitive protective equipment*

IEC 61496-1, *Safety of machinery – Electro-sensitive protective equipment – Part 1: General requirements and tests*

IEC 61508-1:2010, *Functional safety of electrical/electronic/ programmable electronic safety-related systems – Part 1: General requirements*

IEC 61508-4:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 4: Definitions and abbreviations*

IEC 61508-5:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 5: Examples of methods for the determination of safety integrity levels*

IEC 61508-6:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 6: Guidelines on the application of IEC 61508-2 and IEC 61508-3*

IEC 61508-7:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 7: Overview of techniques and measures*

IEC 61511 (all parts), *Functional safety – Safety instrumented systems for the process industry sector*

IEC 61511-1:2016, *Functional safety – Safety instrumented systems for the process industry sector – Part 1: Framework, definitions, system, hardware and application programming requirements*
IEC 61511-1:2016/AMD1:2017
IEC 61511-1:2016/AMD1:2017

IEC 61511-3:2016, *Functional safety – Safety instrumented systems for the process industry sector – Part 3: Guidance for the determination of the required safety integrity levels*

IEC 61649, *Weibull analysis*

IEC 61709:2017, *Electric components – Reliability – Reference conditions for failure rates and stress models for conversion*

IEC 61784-3 (all parts), *Industrial communication networks – Profiles*

IEC 61784-3:~~2016~~2021, *Industrial communication networks – Profiles – Part 3: Functional safety fieldbuses – General rules and profile definitions*

IEC 61800-5-2, *Adjustable speed electrical power drive systems – Part 5-2: Safety requirements – Functional*

IEC 61810 (all parts), *Electromechanical elementary relays*

IEC 62443 (all parts), *Security for industrial automation and control systems*

IEC TS 62443-1-1, *Industrial communication networks – Network and system security – Part 1-1: Terminology, concepts and models*

IEC 62477 (all parts), *Safety requirements for power electronic converter systems and equipment*

IEC 62502, *Analysis techniques for dependability – Event tree analysis (ETA)*

IEC ~~TR~~ TS 63074:~~2019~~2023, *Safety of machinery – Security aspects related to functional safety of safety-related control systems*

IEC TS 63394:2023, *Safety of machinery – Guidelines on functional safety of safety-related control systems*

ISO/IEC Guide 51:2014, *Safety aspects – Guidelines for their inclusion in standards*

ISO/IEC/IEEE 26512, *Systems and software engineering – Requirements for acquirers and suppliers of information for users*

ISO/IEC 27001:~~2013~~2022, ~~*Information technology – Security techniques –*~~ *Information security, cybersecurity and privacy protection – Information security management systems – Requirements*

ISO 4413:2010, *Hydraulic fluid power – General rules and safety requirements for systems and their components*

ISO 4414:2010, *Pneumatic fluid power – General rules and safety requirements for systems and their components*

ISO 11161:2007, *Safety of machinery – Integrated manufacturing systems – Basic requirements*

ISO 13850:2015, *Safety of machinery – Emergency stop function – Principles for design*

ISO 13851:2019, *Safety of machinery – Two-hand control devices – Principles for design and selection*

ISO 13855:2010, *Safety of machinery – Positioning of safeguards with respect to the approach speeds of parts of the human body*

ISO 14118:2017, *Safety of machinery – Prevention of unexpected start-up*

ISO 14119:2013, *Safety of machinery – Interlocking devices associated with guards – Principles for design and selection*

ISO 19973 (all parts), *Pneumatic fluid power – Assessment of component reliability by testing*

ISO TR 22100-1:2015, *Safety of machinery – Relationship with ISO 12100 – Part 1: How ISO 12100 relates to type-B and type-C standards*

ISO TR 22100-4:2018, *Safety of machinery – Relationship with ISO 12100 – Part 4: Guidance to machinery manufacturers for consideration of related IT-security (cyber security) aspects*

ISO 26262 (all parts), *Road vehicles – Functional safety*

EN 50205:2002, *Relays with forcibly guided (mechanically linked) contacts*

EU Guidelines 2010/15/EU (RAPEX)

ISA TR84.00.09:2017, *Cybersecurity Related to the Functional Safety Lifecycle*

MIL-HDBK 217F (Notice 2), Reliability Prediction of Electronic Equipment (28-02-95), Parts Stress Analysis

MIL-HDBK 217F (Notice 2), Reliability Prediction of Electronic Equipment (28-02-95), Appendix A, Parts Count Reliability Prediction

SN 29500 Part 7, Failure Rates of Components, Expected Values for Relays, November 2005

SN 29500 Part 11, Failure Rates of Components, Expected Values for Contactors, April 2015

Failure mode/mechanism distributions FMD-2016, ISBN 978-1-933904-70-2

OREDA Handbook 2015, 6th edition – Volume I and II

EXIDA Safety Equipment Reliability Handbook – 4th edition

EXIDA Electrical & Mechanical Component Reliability Handbook – 3rd Edition

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

SAFETY OF MACHINERY – FUNCTIONAL SAFETY OF SAFETY-RELATED CONTROL SYSTEMS

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) IEC draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). IEC takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, IEC had not received notice of (a) patent(s), which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at https://patents.iec.ch. IEC shall not be held responsible for identifying any or all such patent rights.

This consolidated version of the official IEC Standard and its amendment has been prepared for user convenience.

IEC 62061 edition 2.1 contains the second edition (2021-03) [documents 44/885/FDIS and 44/888/RVD] and its amendment 1 (2024-03) [documents 44/1020/FDIS and 44/1024/RVD].

This Final version does not show where the technical content is modified by amendment 1. A separate Redline version with all changes highlighted is available in this publication.

IEC 62061 has been prepared by IEC technical committee 44: Safety of machinery – Electrotechnical aspects. It is an International Standard.

This second constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

– structure has been changed and contents have been updated to reflect the design process of the safety function,
– standard extended to non-electrical technologies,
– definitions updated to be aligned with IEC 61508-4,
– functional safety plan introduced and configuration management updated (Clause 4),
– requirements on parametrization expanded (Clause 6),
– reference to requirements on security added (Subclause 6.8),
– requirements on periodic testing added (Subclause 6.9),
– various improvements and clarification on architectures and reliability calculations (Clause 6 and Clause 7),
– shift from "SILCL" to "maximum SIL" of a subsystem (Clause 7),
– use cases for software described including requirements (Clause 8),
– requirements on independence for software verification (Clause 8) and validation activities (Clause 9) added,
– new informative annex with examples (Annex G),
– new informative annexes on typical $MTTF_D$ values, diagnostics and calculation methods for the architectures (Annex C, Annex D and Annex H).

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/standardsdev/publications.

The committee has decided that the contents of this document and its amendment will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn, or
- revised.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

As a result of automation, demand for increased production and reduced operator physical effort, Safety-related Control Systems (referred to as SCS) of machines play an increasing role in the achievement of overall machine safety. Furthermore, the SCS themselves increasingly employ complex electronic technology.

IEC 62061 specifies requirements for the design and implementation of safety-related control systems of machinery. This document is machine sector specific within the framework of IEC 61508.

NOTE While IEC 62061 and ISO 13849-1 are using different methodologies for the design of safety related control systems, they intend to achieve the same risk reduction.

This International Standard is intended for use by machinery designers, control system manufacturers and integrators, and others involved in the specification, design and validation of an SCS. It sets out an approach and provides requirements to achieve the necessary performance and facilitates the specification of the safety functions intended to achieve the risk reduction.

This document provides a machine sector specific framework for functional safety of an SCS of machines. It only covers those aspects of the safety lifecycle that are related to safety requirements allocation through to safety validation. Requirements are provided for information for safe use of SCS of machines that can also be relevant to later phases of the lifecycle of an SCS.

There are many situations on machines where SCS are employed as part of safety measures that have been provided to achieve risk reduction. A typical case is the use of an interlocking guard that, when it is opened to allow access to the danger zone, signals the safety related parts of the machine control system to stop hazardous machine operation. In automation, the machine control system that is used to achieve correct operation of the machine process often contributes to safety by mitigating risks associated with hazards arising directly from control system failures. This document gives a methodology and requirements to:

- assign the required safety integrity for each safety function to be implemented by SCS;
- enable the design of the SCS appropriate to the assigned safety (control) function(s);
- integrate safety-related subsystems designed in accordance with other applicable functional safety-related standards (see 6.3.4);
- validate the SCS.

This document is intended to be used within the framework of systematic risk reduction, in conjunction with risk assessment described in ISO 12100. Suggested methodologies for a safety integrity assignment are given in informative Annex A.

SAFETY OF MACHINERY – FUNCTIONAL SAFETY OF SAFETY-RELATED CONTROL SYSTEMS

1 Scope

This International Standard specifies requirements and makes recommendations for the design, integration and validation of safety-related control systems (SCS) for machines. It is applicable to control systems used, either singly or in combination, to carry out safety functions on machines that are not portable by hand while working, including a group of machines working together in a co-ordinated manner.

This document is a machinery sector specific standard within the framework of IEC 61508 (all parts).

The design of complex programmable electronic subsystems or subsystem elements is not within the scope of this document. This is in the scope of IEC 61508 or standards linked to it; see Figure 1.

NOTE 1 Elements such as systems on chip or microcontroller boards are considered complex programmable electronic subsystems.

The main body of this sector standard specifies general requirements for the design, and verification of a safety-related control system intended to be used in high/continuous demand mode.

This document:

– is concerned only with functional safety requirements intended to reduce the risk of hazardous situations;
– is restricted to risks arising directly from the hazards of the machine itself or from a group of machines working together in a co-ordinated manner;

NOTE 2 Requirements to mitigate risks arising from other hazards are provided in relevant sector standards. For example, where a machine(s) is part of a process activity, additional information is available in IEC 61511.

This document does not cover

– electrical hazards arising from the electrical control equipment itself (e.g. electric shock – see IEC 60204-1);
– other safety requirements necessary at the machine level such as safeguarding;
– specific measures for security aspects – see IEC TS 63074.

This document is not intended to limit or inhibit technological advancement.

Figure 1 illustrates the scope of this document.

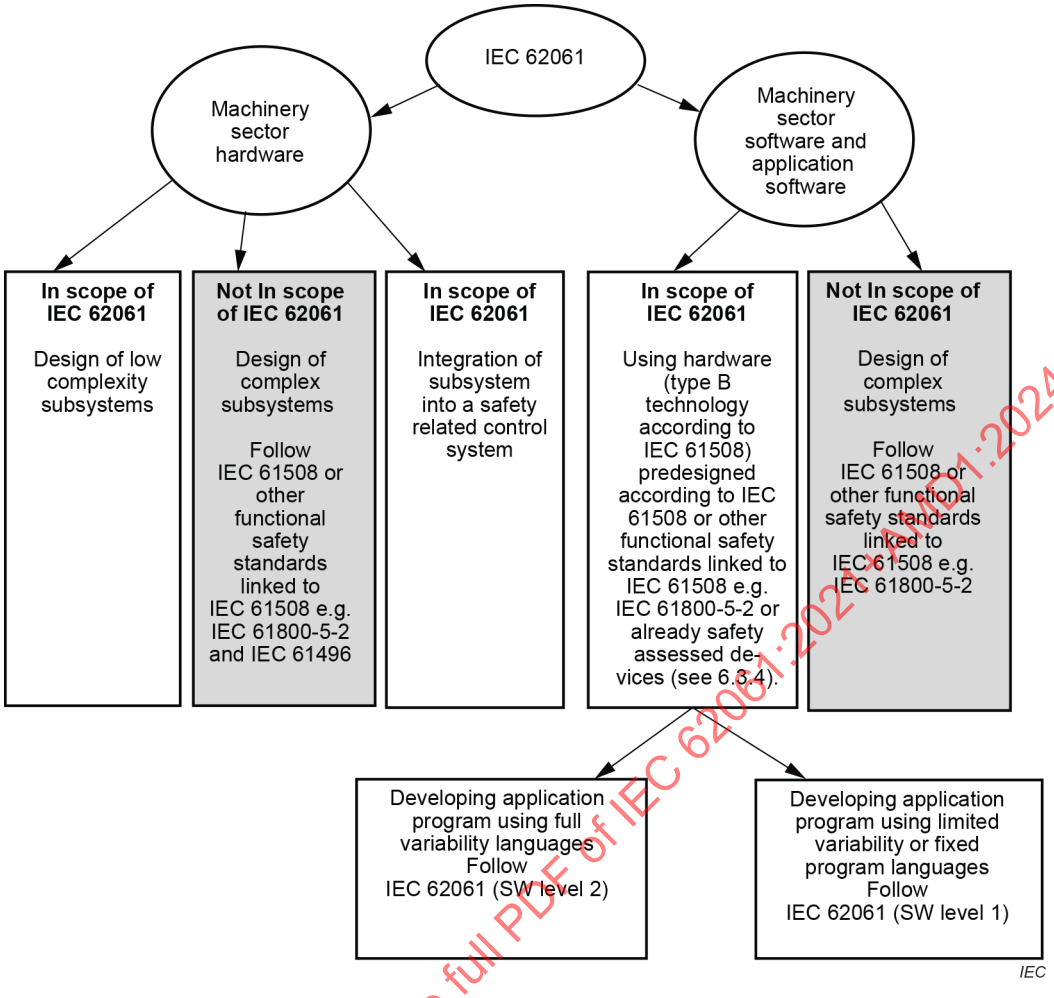


Figure 1 – Scope of this document

2 Normative references

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

IEC 60204-1:2016, *Safety of machinery – Electrical equipment of machines – Part 1: General requirements*

IEC 61000-1-2:2016, *Electromagnetic compatibility (EMC) – Part 1-2: General – Methodology for the achievement of functional safety of electrical and electronic systems including equipment with regard to electromagnetic phenomena*

IEC 61508 (all parts), *Functional safety of electrical/electronic/programmable electronic safety-related systems*

IEC 61508-2:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems*

IEC 61508-3:2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements*

ISO 12100:2010, *Safety of machinery – General principles for design – Risk assessment and risk reduction*

ISO 13849 (all parts), *Safety of machinery – Safety-related parts of control systems*

ISO 13849-1:2015, *Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design*

ISO 13849-2:2012, *Safety of machinery – Safety-related parts of control systems – Part 2: Validation*

3 Terms, definitions and abbreviations

3.1 Alphabetical list of definitions

Terms used throughout IEC 62061 are given in Table 1. Also included are some common abbreviations related to machinery safety.

Table 1 – Terms used in IEC 62061

Term	Definition number
application software	3.2.59
architectural constraint	3.2.46
architecture	3.2.45
average frequency of dangerous failure per hour (PFH)	3.2.29
average probability of dangerous failure on demand (PFD_{avg})	3.2.31
baseline (configuration)	3.2.67
bypass	3.2.17
common cause failure (CCF)	3.2.56
complex component	3.2.8
configuration management	3.2.66
continuous mode	3.2.28
dangerous failure	3.2.52
demand	3.2.25
diagnostic coverage (DC)	3.2.49
diagnostic test interval	3.2.50
embedded software	3.2.60
failure	3.2.51
fault	3.2.33
fault tolerance	3.2.34
full variability language (FVL)	3.2.61
functional safety	3.2.10
hardware fault tolerance (HFT)	3.2.35
hardware safety integrity	3.2.22
harm	3.2.12
hazard	3.2.11
high demand mode	3.2.27
integrator	3.2.13

Term	Definition number
limited variability language (LVL)	3.2.62
low complexity component	3.2.7
low demand mode	3.2.26
machine control system	3.2.2
machinery (machine)	3.2.1
mean repair time (MRT)	3.2.40
mean time to failure ($MTTF$)	3.2.37
mean time to dangerous failure ($MTTF_D$)	3.2.38
mean time to restoration (MTTR)	3.2.39
muting	3.2.16
pre-designed (SCS or subsystem)	3.2.5
probability of dangerous failure on demand (PFD)	3.2.30
process safety time	3.2.41
proof test coverage	3.2.48
proof test	3.2.47
protective measure	3.2.14
random hardware failure	3.2.57
ratio of dangerous failure (RDF)	3.2.55
risk	3.2.15
safe failure	3.2.53
safe failure fraction (SFF)	3.2.54
safe state	3.2.68
safety	3.2.9
safety function	3.2.18
safety integrity	3.2.21
safety integrity level (SIL)	3.2.24
safety-related control system (SCS)	3.2.3
safety-related software	3.2.63
security	3.2.69
(SCS) diagnostic function	3.2.19
(SCS) fault reaction function	3.2.20
subsystem	3.2.4
subsystem element	3.2.6
sub-function	3.2.36
systematic failure	3.2.58
systematic safety integrity	3.2.23
target failure measure	3.2.32
useful lifetime	3.2.42
validation (of the safety function)	3.2.65
verification	3.2.64
well-tried component	3.2.43
well-tried safety principles	3.2.44

3.2 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

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

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

3.2.1
machinery
machine
assembly, fitted with or intended to be fitted with a drive system consisting of linked parts or components, at least one of which moves, and which are joined together for a specific application

Note 1 to entry: The term “machinery” also covers an assembly of machines which, in order to achieve the same end, are arranged and controlled so that they function as an integral whole.

[SOURCE: ISO 12100:2010, 3.1]

3.2.2
machine control system
system that responds to input signals from the machinery and/or from an operator and generates output signals causing the machinery to operate in the desired manner

Note 1 to entry: The machine control system includes input devices and final elements.

[SOURCE: IEC 61508-4:2010, 3.3.3, modified – the term defined has been changed, "process" has been changed to "machinery"]

3.2.3
safety-related control system
SCS
part of the control system of a machine which implements a safety function by one or more subsystems

Note 1 to entry: SCS is similar to SRECS of the previous edition of this document.

3.2.4
subsystem
entity of the top-level architectural design of a safety-related system where a dangerous failure of the subsystem results in dangerous failure of a safety function

Note 1 to entry: This differs from common language where “subsystem” may mean any sub-divided part of an entity, the term “subsystem” is used in this document within a strongly defined hierarchy of terminology: “subsystem” is the first level subdivision of a system. The parts resulting from further subdivision of a subsystem are called “subsystem elements”.

Note 2 to entry: A complete subsystem can be made up from a number of identifiable and separate subsystem elements.

Note 3 to entry: The subsystem specification includes its role in the safety function and its interface with the other subsystems of the SCS.

Note 4 to entry: One subsystem can be part of several safety functions, e.g. the same combination of contactors can be used to de-energise a motor either in the event of detection of a person in a danger zone or also in the event of opening an interlock guard.

[SOURCE: IEC 61508-4:2010, 3.4.4, modified – cross references removed and notes added]

3.2.5
pre-designed SCS or subsystem
SCS or subsystem which meets the relevant requirements of a functional safety standard

3.2.6
subsystem element
part of a subsystem, comprising a single component or any group of components

Note 1 to entry: A subsystem element may comprise hardware and software.

Note 2 to entry: Elements that are not directly necessary for the safety function are not included, but may support it (for example, filters elements, protection against over-voltage).

Note 3 to entry: A subsystem element is the lowest level of detail to consider when ensuring that the requirements of a sub-function are met.

3.2.7
low complexity component/subsystem
component/subsystem in which

– the failure modes are well-defined; and
– the behaviour under fault conditions can be completely defined

Note 1 to entry: Behaviour of the low complexity component / subsystem under fault conditions may be determined by analytical and/or test methods.

Note 2 to entry: Examples of low complexity components / subsystem are limit switches, electro-mechanical relays or contactors.

[SOURCE: IEC 61508-4:2010, 3.4.3, modified – the term defined has been changed, leading to reformulation of text. Example converted into note 2]

3.2.8
complex component / subsystem
component / subsystem in which

– the failure modes are not well-defined; or
– the behaviour under fault conditions cannot be completely defined

3.2.9
safety
freedom from unacceptable risk

[SOURCE: IEC 61508-4:2010, 3.1.11]

3.2.10
functional safety
part of the overall safety of the machine and the machine control system that depends on the correct functioning of the SCS and other risk reduction measures

[SOURCE: IEC 61508-4:2010, 3.1.12, modified – using terms machine, machine control system and SCS]

3.2.11
hazard
potential source of harm

Note 1 to entry: The term hazard can be qualified in order to define its origin or the nature of the expected harm (e.g. electric shock hazard, crushing hazard, cutting hazard, toxic hazard, fire hazard).

[SOURCE: ISO 12100:2010, 3.6, modified – note 1 has been modified and notes 2 and 3 deleted]

3.2.12
harm
injury or damage to the health of people

[SOURCE: ISO/IEC Guide 51:2014, 3.1, modified – without damage to property or the environment]

3.2.13
integrator
entity who designs, manufactures or assembles an integrated manufacturing system and is responsible for the safety strategy, including the protective measures, control interfaces and interconnections of the control system

Note 1 to entry: The integrator may be for example a manufacturer, assembler, engineering company, or entity with the overall responsibility for the machine.

[SOURCE: ISO 11161:2007, 3.10, modified – "provides" has been deleted, last entry in note reformulated]

3.2.14
protective measure
measure intended to achieve risk reduction

[SOURCE: ISO 12100:2010, 3.19, modified – bullet list removed]

3.2.15
risk
combination of the probability of occurrence of harm and the severity of that harm

[SOURCE: ISO/IEC Guide 51:2014, 3.9 modified – note to entry removed]

3.2.16
muting
temporary automatic suspension of a safety function(s)

Note 1 to entry: Other means are used to maintain the safety level.

[SOURCE: ISO 13849-1:2015, 3.1.8, modified – "by the SRP/CS" has been deleted, note added]

3.2.17
bypass
action or facility to prevent all or parts of the SCS functionality from being executed

Note 1 to entry: Examples of bypassing include:

– the input signal is blocked from the trip logic while still presenting the input parameters and alarm to the operator;
– the output signal from the trip logic to a final element is held in the normal state preventing final element operation;
– a physical bypass line is provided around the final element;
– preselected input state (e.g., on/off input) or set is forced by means of an engineering tool (e.g., in the application program).

Note 2 to entry: Other terms are also used to refer to bypassing, such as override, defeat, disable, force, or inhibit or muting.

[SOURCE: IEC 61511-1:2016, 3.2.4, modified – SIS replaced by SCS]

3.2.18
safety function
function implemented by an SCS with a specified integrity level that is intended to maintain the safe condition of the machine or prevent an immediate increase of the risk(s) in respect of a specific hazardous event

Note 1 to entry: This term is used instead of "safety-related control function (SRCF)" of IEC 62061:2015. This definition differs from ISO 12100 because this document addresses risk reduction performed by SCS.

Note 2 to entry: A safety function is typically starting with a detection and evaluation of an 'initiation event' and ending with an output causing a reaction of a 'machine actuator'.

Note 3 to entry: Parts of machine operating function(s), e.g. the reaction of a machine actuator, can also be part of safety function(s).

[SOURCE: IEC 61508-4:2010, 3.5.1, modified – terminology adapted to machinery, other risk reduction measures deleted, example deleted, notes added]

3.2.19
(SCS) diagnostic function
function intended to detect faults in the SCS and initiate a specified fault reaction function when a fault is detected

Note 1 to entry: This function is intended to detect faults that could lead to a dangerous failure of a safety function and initiate a specified fault reaction function.

3.2.20
(SCS) fault reaction function
function that is initiated when a fault within an SCS is detected by the SCS diagnostic function

3.2.21
safety integrity
probability of an SCS or its subsystem satisfactorily performing the required safety function under all stated conditions within a stated period of time

Note 1 to entry: The higher the level of safety integrity of the item, the lower the probability that the item will fail to carry out the required safety function.

Note 2 to entry: Safety integrity comprises hardware safety integrity and systematic safety integrity.

[SOURCE: IEC 61508-4:2010, 3.5.4, modified – terminology adapted to machinery, notes 2, 3, 5 deleted]

3.2.22
hardware safety integrity
part of the safety integrity of an SCS or its subsystems relating to random hardware failures in a dangerous mode of failure

Note 1 to entry: The term relates to failures in a dangerous mode, that is, those failures of a safety-related system that would impair its safety integrity.

Note 2 to entry: Hardware safety integrity includes architectural constraints.

[SOURCE: IEC 61508-4:2010, 3.5.7 – terminology adapted to machinery, note 1 shortened, note 2 added]

3.2.23
systematic safety integrity
part of the safety integrity of an SCS or its subsystems relating to its resistance to systematic failures in a dangerous mode

Note 1 to entry: Systematic safety integrity cannot usually be quantified precisely.

Note 2 to entry: Requirements for systematic safety integrity apply to both hardware and software aspects of an SCS or its subsystems.

[SOURCE: IEC 61508-4:2010, 3.5.6, modified – terminology adapted to machinery, note 1 shortened, note 2 added]

3.2.24
safety integrity level
SIL
discrete level (one out of a possible three) for describing the capability to perform a safety function where safety integrity level three has the highest level of safety integrity and safety integrity level one has the lowest

3.2.25
demand
event that causes the SCS to perform a safety function

Note 1 to entry: Demand mode means that a safety function is only performed on request (demand) in order to transfer the machine into a specified state. The SCS does not influence the machine until there is a demand on the safety function.

Note 2 to entry: Demand rate (DR) or the frequency of demands is one of the main factor that is considered for assessing the demand mode, low or high. For this particular purpose, the demand rate (DR) can be identified with the rate of events, where harm would occur without intervention of the safety function. This rate may be lower than an actual rate of triggering the safety function during operation.

Note 3 to entry: For an emergency stop function, the demand mode is not defined. To determine the achieved SIL, the principle for evaluation of the selected demand mode of the other functions is usually applicable.

3.2.26
low demand mode
mode of operation in which the frequency of demands of a safety function is no greater than one per year

[SOURCE: IEC 61508-4:2010, 3.5.16, modified – low demand extracted from definition of "mode of operation"]

3.2.27
high demand mode
mode of operation in which the frequency of demands of a safety function is greater than one per year

Note 1 to entry: Continuous mode means that a safety function is performed continuously, i.e. the SCS is continuously controlling the machine and a (dangerous) failure of its function can result in a hazard.

Note 2 to entry: The distinction between high demand and continuous mode is relevant for the qualification of diagnostic measures (refer to 7.4.3 and 7.4.4). It is not relevant for target failure measure and SIL assignment.

[SOURCE: IEC 61508-4:2010, 3.5.16, modified – high demand extracted from definition of "mode of operation", notes added]

3.2.28
continuous mode
mode of operation where the safety function retains the machinery in a safe state as a part of normal operation

Note 1 to entry: Continuous mode means that a safety function is performed continuously, i.e. the SCS is continuously controlling the machine and a (dangerous) failure of its function can result in a hazard.

Note 2 to entry: The distinction between high demand and continuous mode is relevant for the qualification of diagnostic measures (refer to 7.4.3 and 7.4.4). It is not relevant for target failure measure and SIL assignment".

[SOURCE: IEC 61508-4:2010, 3.5.16, modified – high demand extracted from definition of "mode of operation", notes added]

3.2.29
average frequency of a dangerous failure per hour
***PFH* or PFH_D**
average frequency of dangerous failure of an SCS to perform a specified safety function over a given period of time

Note 1 to entry: Both terms *PFH* and PFH_D correspond to the probability of dangerous failures per hour (IEC 62061:2005+AMD1:2012+AMD2:2015).

Note 2 to entry: The term “average probability of dangerous failure per hour” is not used in this edition anymore but the acronym *PFH* has been retained but when it is used it means “average frequency of dangerous failure [h]".

[SOURCE: IEC 61508-4:2010, 3.6.19, modified – terminology adapted to machinery, existing notes deleted, new notes added]

3.2.30
probability of dangerous failure on demand
PFD
safety unavailability (see IEC 60050-192) of an SCS to perform the specified safety function when a demand occurs from the machinery or machinery control system

Note 1 to entry: The [instantaneous] unavailability (as per IEC 60050-192) is the probability that an item is not in a state to perform a required function under given conditions at a given instant of time, assuming that the required external resources are provided. It is generally noted by U (t).

Note 2 to entry: The [instantaneous] availability does not depend on the states (running or failed) experienced by the item before it. It characterizes an item which only has to be able to work when it is required to do so, for example, an SCS working in low demand mode.

Note 3 to entry: If periodically tested, the *PFD* of an SCS is, in respect of the specified safety function, represented by a saw tooth curve with a large range of probabilities ranging from low, just after a test, to a maximum just before a test.

[SOURCE: IEC 61508-4:2010, 3.6.17, modified – terminology adapted to machinery]

3.2.31
average probability of dangerous failure on demand
PFD_{avg}
mean unavailability (see IEC 60050-192) of an SCS to perform the specified safety function when a demand occurs from the machinery or machinery control system as an average over time

Note 1 to entry: The mean unavailability over a given time interval [t1, t2] is generally noted by U (t1, t2).

Note 2 to entry: Two kind of failures contribute to *PFD* and PFD_{avg}: the dangerous undetected failures occurred since the last proof test and genuine on demand failures caused by the demands (proof tests and safety demands) themselves. The first one is time dependent and characterized by their dangerous failure rate λ_{DU}(t) whilst the second one is dependent only on the number of demands and is characterized by a probability of failure per demand (denoted by γ).

Note 3 to entry: As genuine on demand failures cannot be detected by tests, it is necessary to identify them and take them into consideration when calculating the target failure measures.

[SOURCE: IEC 61508-4:2010, 3.6.18, modified – terminology adapted to machinery]

3.2.32
target failure measure
intended *PFH* or PFD_{avg} to be achieved to meet a specific safety integrity requirement(s)

Note 1 to entry: Target failure measure is specified in terms of:

– the average probability of a dangerous failure of the safety function on demand, (for a low demand mode of operation);

– the average frequency of a dangerous failure [h^{-1}] (for a high demand mode of operation or a continuous mode of operation).

[SOURCE: IEC 61508-4:2010, 3.5.17, modified – "target probability of dangerous mode failures" changed to "intended PFH or PFD_{avg}", bullet list moved to note 1, existing note deleted]

3.2.33
fault
abnormal condition that may cause a reduction in, or loss of, the capability of an SCS, a subsystem, or a subsystem element to perform a required function

Note 1 to entry: In IEC 60050-192, 192-04-01 a fault of an item is described as inability to perform as required, due to an internal state.

[SOURCE: IEC 61508-4:2010, 3.6.1, modified – terminology adapted to machinery, note shortened]

3.2.34
fault tolerance
ability of an SCS, a subsystem, or subsystem element to continue to perform a required function in the presence of faults or failures

[SOURCE: IEC 61508-4:2010, 3.6.3, modified – terminology adapted to machinery]

3.2.35
hardware fault tolerance
HFT
property of a subsystem to potentially lose the safety function upon at least N+1 faults

Note 1 to entry: A hardware fault tolerance of N means that N+1 faults of a subsystem could cause a loss of the safety function.

[SOURCE: IEC 61508-2:2010, derived from 7.4.4.1.1]

3.2.36
sub-function
part of a safety function whose failure can result in a failure of the safety function

Note 1 to entry: In this document, a safety function can be seen as a logical AND of the sub-functions.

3.2.37
mean time to failure
MTTF
expectation of the mean time to failure

Note 1 to entry: $MTTF$ is normally expressed as an average value of expectation of the time to failure.

[SOURCE: IEC 60050-192, 192-05-11, modified – note added and original notes removed]

3.2.38
mean time to dangerous failure
***MTTF*$_D$**
expectation of the mean time to dangerous failure

[SOURCE: Definition derived from IEC 60050-192, 192-05-11, modified – restricted to dangerous failures]

3.2.39
mean time to restoration
MTTR
expected time to achieve restoration after a fault has occurred in a safety function.

Note 1 to entry: MTTR encompasses:

- the time to detect the failure (a); and
- the time spent before starting the repair (b); and
- the effective time to repair (c); and
- the time before the component is put back into operation (d).

The start time for (b) is the end of (a); the start time for (c) is the end of (b); the start time for (d) is the end of (c).

Note 2 to entry: During this time the machine can continue to operate

[SOURCE: IEC 61508-4:2010, 3.6.21, modified – terminology adapted to machinery and more details added to definition]

3.2.40
mean repair time
MRT
mean repair time after a fault has been detected in a safety function and machine continues to operate

Note 1 to entry: MRT encompasses:

- the time spent before starting the repair (b); and
- the effective time to repair (c); and
- the time before the component is put back into operation (d).

Note 2 to entry: Depending on the type of detected fault and the fault reaction, the numerical values for MRT and MTTR can be different.

[SOURCE: IEC 61508-4:2010, 3.6.22, modified – terminology adapted to machinery and more details added to definition, note 1 made similar to 3.2.39, note 2 added]

3.2.41
process safety time
period of time between a failure, that has the potential to give rise to a hazardous event, occurring in the machinery or machinery control system and the time by which action has to be completed in the machinery to prevent the hazardous event occurring

Note 1 to entry: It is foreseen that the safety function detects the failure and completes its action soon enough to prevent the hazardous event taking into account any process lag (e.g. stopping times).

[SOURCE: IEC 61508-4:2010, 3.6.20 modified – terminology adapted to machinery, note 1 added]

3.2.42
useful lifetime
minimum elapsed time between the installation of the SCS or subsystem or subsystem element and the point in time when component failure rates of the SCS or subsystem or subsystem element can no longer be predicted, with any accuracy

Note 1 to entry: Typically it will be 20 years or less unless the manufacturers of the SCS and its subsystems can justify a longer lifetime by providing evidence, based on calculations, showing that reliability data is valid for the longer lifetime.

[SOURCE: IEC 61131-6:2012, 3.57, modified – terminology adapted to machinery, note 1 added, example deleted]

3.2.43
well-tried component
for a safety-related application, component for a safety-related application which has been either

a) widely used in the past with successful results in similar safety-related applications as given as well-tried components in the informative annexes of ISO 13849-2, or

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

Note 1 to entry: ISO 13849-2 lists a variety of components and the conditions for specific technologies under which the component can be considered well-tried.

Note 2 to entry: Newly developed components may be considered as equivalent to “well-tried” if they fulfil the conditions of b).

Note 3 to entry: The decision to accept a particular component as being “well-tried” depends on the application, e.g. owing to the environmental influences and can be impacted by product or manufacturer changes.

Note 4 to entry: Complex electronic components (e.g. PLC, microprocessor, application-specific integrated circuit) cannot be considered as equivalent to “well tried”.

Note 5 to entry: A well-tried component is not a proven in use component.

3.2.44
well-tried safety principles
principles that have proved effective in the design or integration of safety-related control systems in the past, to avoid or control critical faults or failures which can influence the performance of a safety function

Note 1 to entry: Newly developed safety principles can be considered as equivalent to “well-tried” if they are verified using principles which demonstrate their suitability and reliability for safety-related applications.

Note 2 to entry: Well-tried safety principles are effective not only against random hardware failures, but also against systematic failures which may creep into the product at some point in the course of the product life cycle, e.g. faults arising during product design, integration, modification or deterioration.

Note 3 to entry: Tables A.2, B.2, C.2 and D.2 in the informative annexes of ISO 13849-2:2012 address well-tried safety principles for different technologies.

[SOURCE: Definition derived from ISO 13849-1:2015]

3.2.45
architecture
specific configuration of hardware and software elements in an SCS

[SOURCE: IEC 61508-4:2010, 3.3.4, modified – terminology adapted to machinery]

3.2.46
architectural constraint
set of architectural requirements that limit the SIL that can be claimed for a subsystem

3.2.47
proof test
periodic test that can detect dangerous undetected faults and degradation in an SCS and its subsystems so that, if necessary, the relevant parts of the SCS and its subsystems can be restored to an “as new” condition or as close as practical to this condition

Note 1 to entry: A proof test is intended to confirm that relevant parts of an SCS are in a condition that assures the specified safety integrity.

Note 2 to entry: The effectiveness of the proof test will be dependent both on failure coverage and repair effectiveness. In practice, detecting 100 % of the degradation that could lead to the hidden dangerous failures later on is not easily achieved. For complex elements or safety features that are difficult to verify, a proof test coverage of 100 % cannot be usually obtained.

[SOURCE: IEC 61508-4:2010, 3.8.5, modified – terminology adapted to machinery, notes 1, 3, 4 deleted, new note 1 added, note 2 shortened]

3.2.48
proof test coverage
term given to the percentage of dangerous undetected failures that are detected by a defined proof test procedure

Note 1 to entry: It measures the effectiveness of a proof test and ranges from 0 % to 100 % (perfect proof-test).

Note 2 to entry: For example, a PTC of 95 % states that 95 % of all possible undetected failures will be detected during the proof test. It doesn't include aging or degradation not directly related to the safety function failure.

Note 3 to entry: The PTC can be estimated by the means of Failure Mode and Effects Analysis (FMEA) in conjunction with engineering judgement based on sound evidence.

3.2.49
diagnostic coverage
DC
fraction of dangerous failures detected by automatic on-line diagnostic tests

Note 1 to entry: The fraction of dangerous failures is computed by using the dangerous failure rates associated with the detected dangerous failures divided by the total rate of dangerous failures.

Note 2 to entry: The dangerous failure diagnostic coverage is computed using the following equation, where DC is the diagnostic coverage, λ_{DD} is the detected dangerous failure rate and λ_{Dtotal} is the total dangerous failure rate:

$$DC = \frac{\sum \lambda_{DD}}{\sum \lambda_{Dtotal}} \quad (1)$$

Note 3 to entry: This definition is applicable providing the individual components have constant failure rates.

[SOURCE: IEC 61508-4:2010, 3.8.6, modified – part of the definition has been moved to a note to entry]

3.2.50
diagnostic test interval
interval between on-line tests to detect faults in a subsystem that has a specified diagnostic coverage

[SOURCE: IEC 61508-4:2010, 3.8.7, modified – replacing safety-related system by subsystem]

3.2.51
failure
termination of the ability of an item (SCS, a subsystem or a subsystem element) to perform a required function

Note 1 to entry: Failures are either random (in hardware) or systematic (in hardware or software).

Note 2 to entry: After a failure, the item has a fault.

Note 3 to entry: "Failure" is an event, as distinguished from "fault", which is a state.

Note 4 to entry: The concept of failure as defined does not apply to items consisting of software only.

[SOURCE: IEC 61508-4:2010, 3.6.4, modified and ISO 12100-1:2010, 3.32]

3.2.52
dangerous failure
failure of an SCS, a subsystem, or a subsystem element that plays a part in implementing the safety function that:

a) prevents a safety function from operating when required (demand mode) or causes a safety function to fail (continuous mode) such that the machine is put into a hazardous or potentially hazardous state; or

b) decreases the probability that the safety function operates correctly when required

[SOURCE: IEC 61508-4:2010, 3.6.7, modified – Terminology adapted to machinery]

3.2.53
safe failure
failure of an SCS, a subsystem, or a subsystem element that plays a part in implementing the safety function that:

a) results in the spurious operation of the safety function to put the machine (or part thereof) into a safe state or maintain a safe state; or

b) increases the probability of the spurious operation of the safety function to put the machine (or part thereof) into a safe state or maintain a safe state

[SOURCE: IEC 61508-4:2010, 3.6.8, modified – terminology adapted to machinery]

3.2.54
safe failure fraction
SFF
fraction of the overall failure rate of a subsystem that does not result in a dangerous failure

Note 1 to entry: The diagnostic coverage (if any) of each subsystem in SCS is taken into account in the calculation of the probability of random hardware failures. The safe failure fraction is taken into account when determining the architectural constraints on hardware safety integrity (see 7.4).

Note 2 to entry: "No effect failures" and "no part failures" (see IEC 61508-4) is not used for SFF calculations.

3.2.55
ratio of dangerous failure
RDF
fraction of the overall failure rate of an element that can result in a dangerous failure

3.2.56
common cause failure
CCF
failure, that is the result of one or more events, causing concurrent failures of two or more separate channels in a multiple channel subsystem, leading to failure of a safety function

[SOURCE: IEC 61508-4:2010, 3.6.10, modified – system failure replaced by failure of a safety function]

3.2.57
random hardware failure
failure occurring at a random time, which results from one or more of the possible degradation mechanisms in the hardware

[SOURCE: IEC 61508-4:2010, 3.6.5, modified – notes removed]

3.2.58
systematic failure
failure, related in a deterministic way to a certain cause, which can only be eliminated by a modification of the design or of the manufacturing process, operational procedures, documentation or other relevant factors

Note 1 to entry: Corrective maintenance without modification will usually not eliminate the failure cause.

Note 2 to entry: A systematic failure can be induced by simulating the failure cause.

Note 3 to entry: Examples of causes of systematic failures include human error in

- the safety requirements specification;
- the design, manufacture, installation and/or operation of the hardware;
- the design and/or implementation of the software.

[SOURCE: IEC 61508-4:2010, 3.6.6, modified – note 3 slightly changed, note 4 removed]

3.2.59
application software
software specific to the application, that is implemented by the designer of the SCS, generally containing logic sequences, limits and expressions that control the appropriate input, output, calculations, and decisions necessary to meet the SCS functional requirements

3.2.60
embedded software
software, supplied as part of a pre-designed subsystem, that is not intended to be modified and that relates to the functioning of, and services provided by, the SCS or subsystem, as opposed to the application software

Note 1 to entry: Firmware and system software are examples of embedded software.

3.2.61
full variability language
FVL
type of language that provides the capability to implement a wide variety of functions and applications

Note 1 to entry: Typical example of systems using FVL are general-purpose computers.

Note 2 to entry: FVL is normally found in embedded software and is rarely used in application software.

Note 3 to entry: FVL examples include: Ada, C, Pascal, Instruction List, assembler languages, C++, Java, SQL.

[SOURCE: IEC 61511-1:2016, 3.2.75.3, modified – first part of definition supressed and link to process sector deleted]

3.2.62
limited variability language
LVL
type of language that provides the capability to combine predefined, application specific, library functions to implement the safety requirements specifications

Note 1 to entry: A LVL provides a close functional correspondence with the functions required to achieve the application.

Note 2 to entry: Typical examples of LVL are given in IEC 61131-3. They include ladder diagram, function block diagram and sequential function chart. Instruction lists and structured text are not considered to be LVL.

Note 3 to entry: Typical example of systems using LVL: Programmable Logic Controller (PLC) configured for machine control.

[SOURCE: IEC 61511-1:2016, 3.2.75.2, modified – note 1 turned into definition, note 2 deleted, note 3 replaced]

3.2.63
safety-related software
software that is used to implement safety functions in a safety-related system

3.2.64
verification
confirmation by examination (e.g. tests, analysis) that the SCS, its subsystems or subsystem elements meet the requirements set by the relevant specification

EXAMPLE: Verification activities include

- reviews on outputs (documents from all phases) to ensure compliance with the objectives and requirements of the phase, taking into account the specific inputs to that phase;
- design reviews;
- tests performed on the designed products to ensure that they perform according to their specification;
- integration tests performed where different parts of a system are put together in a step-by-step manner and by the performance of environmental tests to ensure that all the parts work together in the specified manner.

[SOURCE: IEC 61508-4:2010, 3.8.1, modified – terminology adapted to machinery, note deleted]

3.2.65
validation (of the safety function)
confirmation by examination (e.g. tests, analysis) that the SCS meets the functional safety requirements of the specific application

[SOURCE: IEC 61508-4:2010, 3.8.2, modified – terminology adapted to machinery, notes deleted]

3.2.66
configuration management
discipline of identifying the components of an evolving system for the purposes of controlling changes to those components and maintaining continuity and traceability throughout the lifecycle

[SOURCE: IEC 61508-4:2010, 3.7.3, modified – note removed]

3.2.67
baseline (configuration)
well-defined set of elements (hardware, software, documentation, tests, etc.) of an SCS at a specific point in time.

Note 1 to entry: A baseline serves as a basis for verification, validation, modification and changes.

Note 2 to entry: If an element is changed, the status of the baseline is intermediate until a new baseline is defined.

3.2.68
safe state
state of the machine when safety is achieved

Note 1 to entry: The safe state doesn't include the restoration of initial equipment failures.

[SOURCE: IEC 61508-4:2010, 3.1.13, modified – terminology adapted to machinery, original note deleted, note 1 added]

3.2.69
security
1) measures taken to protect a system
2) condition of a system that results from the establishment and maintenance of measures to protect the system
3) condition of system resources being free from unauthorized access and from unauthorized or accidental change, destruction, or loss

4) capability of a computer-based system to provide adequate confidence that unauthorized persons and systems can neither modify the software and its data nor gain access to the system functions, and yet to ensure that this is not denied to authorized persons and systems

5) prevention of illegal or unwanted penetration of, or interference with the proper and intended operation of an industrial automation and control system

Note 1 to entry: Measures can be controls related to physical security (controlling physical access to computing assets) or logical security (capability to login to a given system and application).

[SOURCE: IEC TS 62443-1-1:2009, 3.2.99]

3.3 Abbreviations

Abbreviations used in this document are shown in Table 2.

Table 2 – Abbreviations used in IEC 62061

CCF	Common Cause Failure(s)
DC	Diagnostic Coverage
EMC	Electromagnetic Compatibility
FVL	Full Variability Language
I/O	Input/Output
LVL	Limited Variability Language
HFT	Hardware Fault Tolerance
HW	Hardware
PFH, PFH_D	average frequency of dangerous failure per Hour
MRT	Mean Repair Time
MTTF	Mean Time To Failure
$MTTF_D$	Mean Time to Dangerous Failure
MTTR	Mean Time To Restoration
PFD	probability of dangerous failure on demand
PFD_{avg}	average probability of dangerous failure on demand
PL	Performance Level
PLC	Programmable Logic Controller
RDF	Ratio of Dangerous Failure
SFF	Safe Failure Fraction
SIL	Safety Integrity Level
SCS	Safety-Related Control System
SRS	Safety Requirements Specification
SW	Software

4 Design process of an SCS and management of functional safety

4.1 Objective

The objective of Clause 4 is to describe the design process and the tasks that have to be completed to realize each safety function performed by the related part of the control system for a given machine.

4.2 Design process

If as a result of the risk assessment of the whole machine according to ISO 12100 (see Figure 2), a need for risk reduction has been identified and if certain selected risk reduction measures depend on the control system, corresponding safety functions have to be specified.

NOTE 1 Examples of safety functions are given in Annex G.

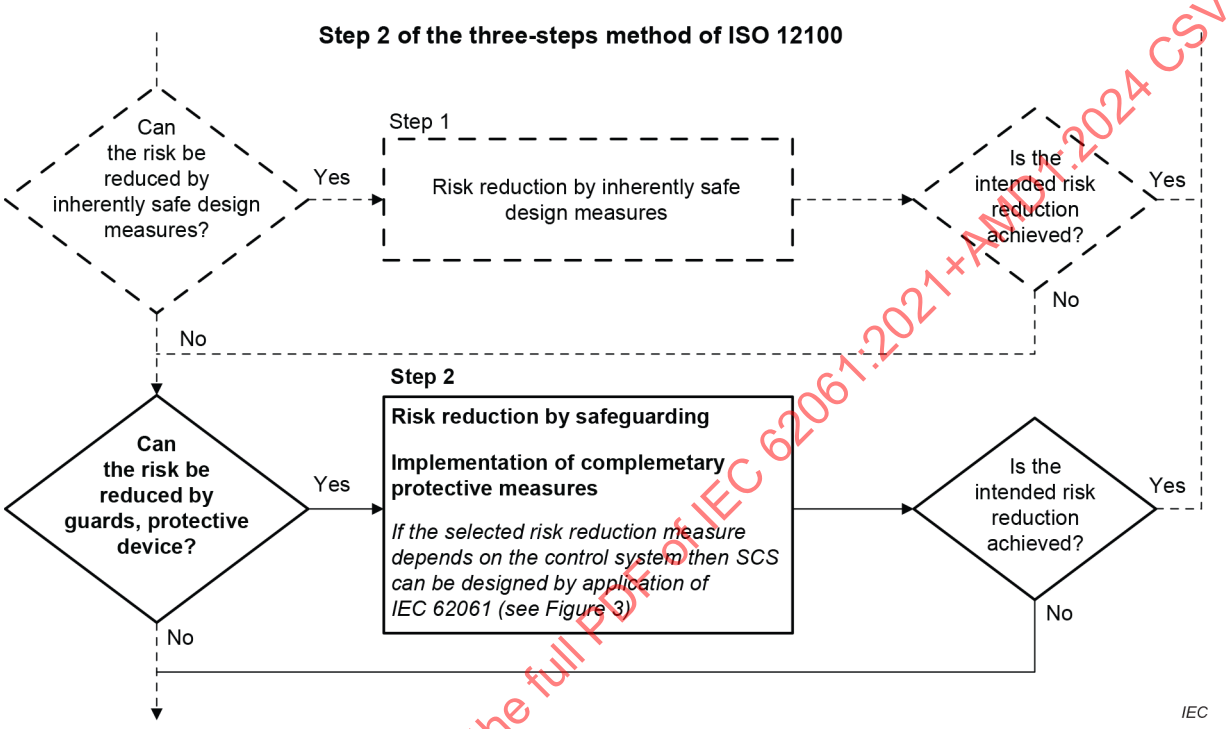
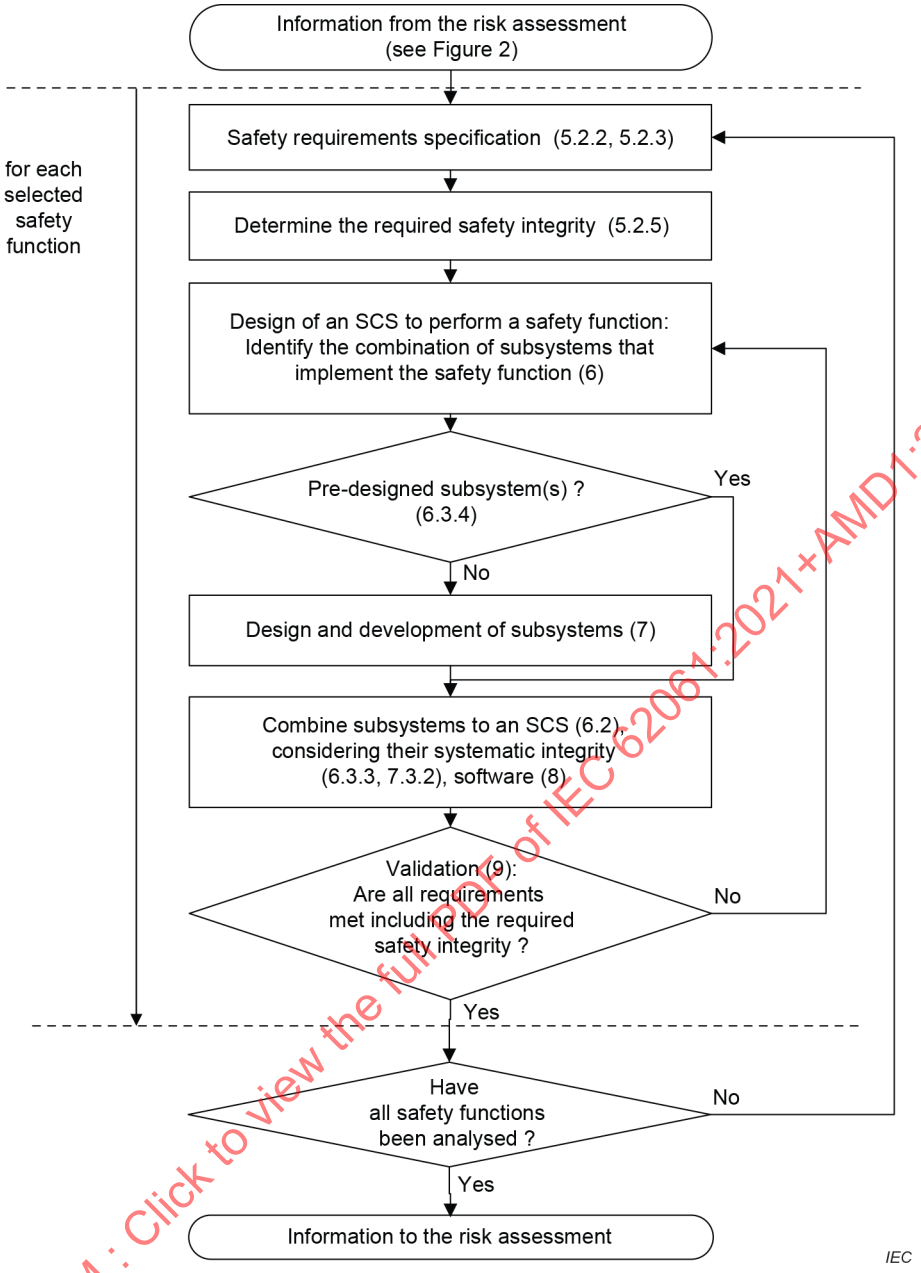


Figure 2 – Integration within the risk reduction process of ISO 12100 (extract)

NOTE 2 Figure 2 shows where the SCS contributes to the risk reduction process of ISO 12100: Step 2. The SCS supports the combined protective measures by the implementation of safety functions. ISO 12100 also provides general design rules for the machine which are applicable for the design of the SCS (see 6.2.11 and 6.2.12 of ISO 12100:2010).

The design process (see Figure 3) of each safety function implemented by a safety-related control system (SCS) shall include at least the safety function specification (see Clause 5) and the safety-related control system design (see Clause 6) and the associated verification and validation activities.



NOTE Each step described in the process flow diagram includes also verification activities.

Figure 3 – Iterative process for design of the safety-related control system

The realization of a safety function following the determined required safety integrity shall either be done by

– using an already developed SCS that meets the required safety integrity, or
– designing a new SCS using pre-designed subsystems according to Clause 6 or designing new subsystems according to Clause 7, or a combination of both.

If additional design considerations for software are necessary, Clause 8 applies.

A safety function can be implemented by one or more subsystem(s) of a safety-related control system (SCS), and several safety functions can share one or more subsystem(s) (e.g. a logic unit, power control element(s)), see examples in Figure 4. A control system can be subdivided into a safety-related part and a non-safety-related part. It is possible that one subsystem, which is involved in the implementation of safety functions, is also involved in the implementation of control functions. The designer may use any of the technologies available, singly or in combination.

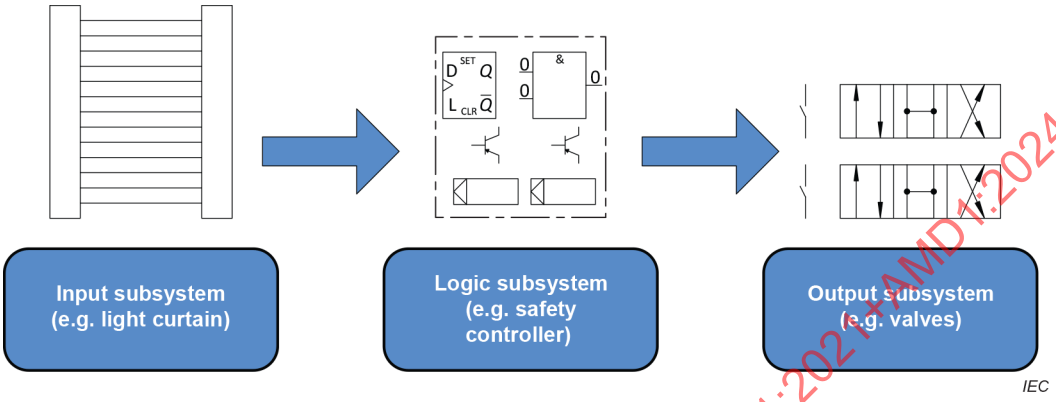


Figure 4 – Example of a combination of subsystems as one SCS

4.3 Management of functional safety using a functional safety plan

This subclause specifies management and technical activities that are necessary for the achievement of the required functional safety of the SCS.

NOTE 1 For further information, see IEC 61508-1:2010, Clause 6.

A functional safety plan shall be drawn up and documented for each SCS design project, and shall be updated as necessary. The functional safety plan is intended to provide measures for preventing incorrect specification, implementation, or modification issues.

The functional safety plan shall identify the relevant activities (see Figure 3) and shall be adapted to the project. See examples in Annex I.

NOTE 2 The functional safety plan can be part of a global machine design plan.

NOTE 3 The content of the functional safety plan depends upon the specific circumstances, which can include:

– size of project;
– degree of complexity;
– degree of novelty of design and technology;
– degree of standardization of design features;
– possible consequence(s) in the event of failure.

In particular, the functional safety plan shall:

a) identify the relevant activities specified in Clauses 5 to 9 and details of when they shall take place;
b) describe the policy and strategy to fulfil the specified functional safety requirements;
c) describe the strategy to achieve functional safety for the application software, results of a development, integration, verification and validation;
d) identify persons, departments or other units and resources that are responsible for carrying out and reviewing each of the activities specified in Clauses 5 to 9.

NOTE 4 The level of appropriate competency of the involved persons (i.e. training, technical knowledge, experience and qualifications) are taken into account. The appropriateness of competence is considered in relation to the particular application, taking into account all relevant factors including:

a) the responsibilities of the person;

b) the level of supervision required;

c) the potential consequences in the event of failure of the SCS;

d) the safety integrity levels of the SCS;

e) the novelty of the design, design procedures or application;

f) previous experience and its relevance to the specific duties to be performed and the technology being employed;

g) the type of competence appropriate to the circumstances (for example qualifications, experience, relevant training and subsequent practice, and leadership and decision-making abilities);

h) engineering knowledge appropriate to the application area and to the technology;

i) safety engineering knowledge appropriate to the technology;

j) knowledge of the legal and safety regulatory framework;

k) relevance of qualifications to specific activities to be performed.

e) identify or establish the procedures and resources to record and maintain information relevant to the functional safety of an SCS;

NOTE 5 The following are considered:

– the results of the hazard identification and risk assessment;

– the equipment used for safety-related functions together with its safety requirements;

– the organization responsible for maintaining functional safety;

– the procedures necessary to achieve and maintain functional safety (including SCS modifications).

f) describe the strategy for configuration management (see 4.4) taking into account relevant organizational issues, such as authorized persons and internal structures of the organization;

g) describe the strategy for modification (see 4.5);

h) establish a verification plan that shall include:

– details of when the verification shall take place;

– details of the persons, departments or units who shall carry out the verification;

– the selection of verification strategies and techniques;

– the selection and utilization of test equipment;

– the selection of verification activities;

– acceptance criteria; and

– the means to be used for the evaluation of verification results;

i) establish a validation plan comprising:

– results of previous verification;

– details of when the validation shall take place;

– identification of the relevant modes of operation of the machine (e.g. normal operation, setting);

– requirements against which the SCS shall be validated;

– the technical strategy for validation, for example analytical methods or statistical tests;

– acceptance criteria; and

– actions to be taken in the event of failure to meet the acceptance criteria.

NOTE 6 The validation plan indicates whether the SCS and its subsystems are to be subject to routine testing, type testing and/or sample testing.

4.4 Configuration management

The main operational aspects of configuration management are

– **identification** of the structure of the SCS, identifies e.g. system, subsystems, functions, function blocks, management documents, tools for creating a baseline;

– **controlling** of the release of an element created during each lifecycle phase at a specific point in time;

– **recording** and **reporting** of the status of each element which is and/or will be part of a baseline;

– **audit** and **review** of all elements and maintaining consistency among all elements of a baseline.

Procedures shall be developed for configuration management of the SCS during the overall, SCS system and software safety lifecycle phases, including in particular:

a) the point, in respect of specific phases, at which formal configuration control is to be implemented;

b) the procedures to be used for uniquely identifying all constituent parts of hardware and software;

c) the procedures for preventing unauthorized items from entering service.

The configuration management procedures shall be implemented in accordance with the functional safety plan (see 4.3).

The procedures for an appropriate change-control-process shall consider the requirements of procedures for defining a unique baseline of each version of the SCS.

4.5 Modification

If a modification is to be implemented, then relevant activities shall be identified specifically and an action plan shall be prepared and documented before carrying out any modification.

NOTE 1 The request for a modification can arise from, for example:

– safety requirements specification changed;

– conditions of actual use;

– incident/accident experience;

– change of material processed;

– obsolescence;

– modifications of the machine or of its operating modes.

NOTE 2 Interventions (e.g. adjustment, setting, repairs) on the SCS made in accordance with the information for use or instruction manual for the SCS are not considered to be a modification in the context of this subclause.

The reason(s) for the request for a modification shall be documented.

The effect of the requested modification shall be analysed to establish the effect on the safety function.

The modification impact analysis and the effect on the functional safety of the SCS shall be documented.

All accepted modifications that have an effect on the SCS shall initiate a return to an appropriate design phase for its hardware and/or for its software (e.g. specification, design, integration, installation, commissioning, and validation). All subsequent phases and management procedures shall then be carried out in accordance with the procedures specified for the specific phases in this document. All relevant documents shall be revised, amended and reissued accordingly.

5 Specification of a safety function

5.1 Objective

This clause sets out the procedures to specify the requirements of safety function(s) to be implemented by the SCS.

5.2 Safety requirements specification (SRS)

5.2.1 General

Each safety function shall be specified by:

– functional requirements specification (see 5.2.3);
– safety integrity requirements specification (see 5.2.5)

and these shall be documented in the safety requirements specification (SRS).

Where a product standard specifies the safety requirements for the design of an SCS or subsystem (e.g. ISO 13851 for two-hand control devices), these should be considered.

5.2.2 Information to be available

The following information shall be used to produce both the functional requirements specification and safety integrity requirements specification of SCS:

– results of the risk assessment for the machine including all safety functions determined to be necessary for the risk reduction process for each specific hazard;
– machine operating characteristics, including:
  - modes of operation of machine,
  - cycle time,
  - response time performance,
  - environmental conditions,
  - interaction of person(s) with the machine (e.g. repairing, setting, cleaning);
– all information relevant to the safety function(s) which can have an influence on the SCS design including, for example:
  - a description of the behaviour of the machine that a safety function is intended to achieve or to prevent;
  - all interfaces between the safety functions, and between safety functions and any other function (either within or outside the machine);
  - required fault reaction functions of the safety function.

NOTE Some of the information might not be available or sufficiently defined before starting the iterative design process of SCS, so the SCS safety requirements specifications can be required to be updated during the design process.

5.2.3 Functional requirements specification

The functional requirements specification shall describe details of each safety function to be performed including as applicable:

- a description of each safety function;
- the condition(s) (e.g. operating mode) of the machine in which the safety function shall be active, disabled, configured or parameterized;
- the priority of those functions that can be simultaneously active and that can cause conflicting action;
- the reset of a safety function;
- the frequency of operation of each safety function (rate of operating cycles, duty cycle);
- demand mode of operation;

  NOTE 1 For definitions refer to 3.2.26, 3.2.27, 3.2.28.
- the required response time of each safety function;
- the interface(s) of the safety functions to other machine functions;

  NOTE 2 This could include a description of methods intended to give status information to users of the machinery.
- a description of fault reaction function(s) and any constraints on, for example, re-starting or continued operation of the machine in cases where the initial fault reaction is to stop the machine;
- tests and any associated facilities (e.g. test equipment, test access ports);
- a description of the operating environment (e.g. electromagnetic immunity, temperature, humidity, dust, chemical substances, mechanical vibration and shock);

  NOTE 3 The specification of the electromagnetic environmental condition is within the scope of IEC 61000-1-2. The electromagnetic environment is defined as the totality of electromagnetic phenomena existing at a particular location. These phenomena can vary over time.

  The electromagnetic environment is influenced by, for example:
  - fixed and moving sources of electromagnetic energy,
  - low, medium and high voltage equipment,
  - control, signalling, communication and power systems,
  - intentional radiators,
  - physical processes (e.g. atmospheric discharges, switching actions),
  - random or infrequent transients,

  which all can produce disturbances that adversely impact the safety-related system or element under consideration.
- rate of operating cycles, duty cycle, and/or utilisation category, for devices intended for use in the safety function;

  NOTE 4 The duty cycle of subsystems or subsystem elements can be higher than required for the safety function, e.g. when used also for non-safety-related machine functions (the total number of cycles is to be considered).
- other specific requirements which can impact functional safety.

5.2.4 Estimation of demand mode of operation

The demand mode of operation shall be estimated by applying the respective definitions. While low demand mode operation is possible for a safety function, this document concentrates on high demand and continuous mode. When demand rate is estimated to be low, a high demand mode can be assumed by activation of the safety function at least once per year. Then apply this document for the design. This is a straightforward application of the definition and shown in Figure 5 as a workflow.

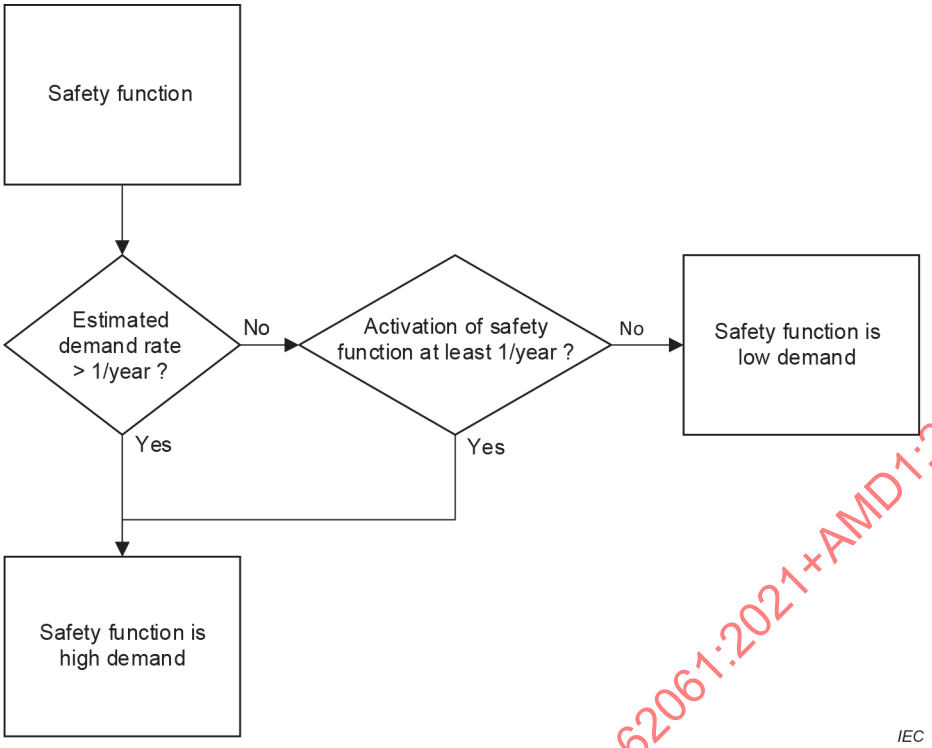


Figure 5 – By activating a low demand safety function at least once per year it can be assumed to be high demand

5.2.5 Safety integrity requirements specification

The safety integrity requirements for each safety function shall be derived from the risk assessment to ensure the necessary risk reduction can be achieved. In this document, a safety integrity requirement is expressed as a target failure measure for the PFH.

The required safety integrity for each safety function to be carried out by an SCS shall be specified in terms of SIL according to Table 3 and documented.

Table 3 – SIL and limits of PFH values

SIL	Limits of PFH values (1/h)
1	$< 10^{-5}$
2	$< 10^{-6}$
3	$< 10^{-7}$

The determination of the required safety integrity is the result of the risk assessment and refers to the amount of the risk reduction to be carried out by the SCS. Examples of a methodology are given in Annex A.

NOTE 1 Where a product standard specifies a required SIL for a safety function then this takes precedence over Annex A.

NOTE 2 Further guidance on relationship between risk assessment according to ISO 12100 and product standards is provided in ISO TR 22100-1.

6 Design of an SCS

6.1 General

The SCS shall be designed in accordance with the safety requirements specification (see 5.2), using one or several subsystems by:

– selection of subsystems (see 6.2, 6.3 and Clause 7);
– determining the safety integrity (see 6.4);
– complying to the requirements of the systematic safety integrity of the SCS (see 6.5), including, where applicable, electromagnetic immunity (see 6.6), security (see 6.8), periodic testing (see 6.9) and, software (see 6.7 and Clause 8).

6.2 Subsystem architecture based on top down decomposition

The following Clause 6 describes the design process of an SCS. An SCS can include:

– one or several pre-designed subsystem(s), and/or
– one or several subsystem(s) developed according to this document, based on subsystem element(s) (see Clause 7).

NOTE 1 The designer of a pre-designed subsystem can be a manufacturer of the machine or a device manufacturer.

NOTE 2 The relevant safety integrity characteristic values come from the designer of pre-designed subsystem.

6.3 Basic methodology – Use of subsystem

6.3.1 General

Each safety function identified in the risk reduction process (see Clause 4) is performed by an SCS consisting of one or several subsystems. A failure of any subsystem will result in the loss of the whole safety function. Subclause 6.2 describes the principle of this allocation task.

Where an SCS or part of an SCS (i.e. its subsystem(s)) is to implement both safety functions and other functions, then all its hardware and software shall be treated as safety-related unless it can be shown that the implementation of the safety functions and other functions is sufficiently independent (i.e. that the normal operation or failure of any other functions do not affect the safety functions).

NOTE 1 Sufficient independence of implementation is established by showing that the probability of a dependent failure between the non-safety and safety-related parts is equivalent to that of the safety integrity level of the SCS. IEC 61508-3:2010, Annex F describes techniques for achieving non-interference between software elements.

For an SCS or its subsystems that implements safety functions of different safety integrity levels, its hardware and software shall be treated as requiring the highest safety integrity level unless it can be shown that the implementation of the safety functions of the different safety integrity levels is sufficiently independent.

NOTE 2 Sufficient independence of implementation is established by showing that the probability of a dependent failure between the non-safety and safety-related parts is sufficiently low in comparison with the highest safety integrity level associated with the safety functions involved.

Where digital data communication is used as a part of an SCS implementation, it shall satisfy the relevant requirements of IEC 61508-2:2010, 7.4.11 (which refers to IEC 61784-3 (all parts) for functional safety fieldbuses) in accordance with the SIL target(s) of the safety function(s).

6.3.2 SCS decomposition

Each safety function shall be decomposed to a structure of sub-function(s). The decomposition process shall lead to a structure of sub-functions that fully describes the functional and integrity requirements of the SCS. This process should be applied down to that level that permits the functional and integrity requirements determined for each sub-function to be allocated to a single subsystem.

Figure 6 shows examples of typical decompositions starting with a detection and evaluation of an ‘initiation event’ and is ending with an output causing a reaction of a ‘machine actuator’.

For each sub-function the following shall be specified:

– the safety requirements (functional and integrity), and
– inputs and outputs of each sub-function.

NOTE 1 The inputs and outputs of each sub-function are the information that is transferred, for example speed, position, mode of operation, etc.

NOTE 2 The sub-functions can have associated diagnostic functions (see 7.4.3.3, diagnostic coverage).

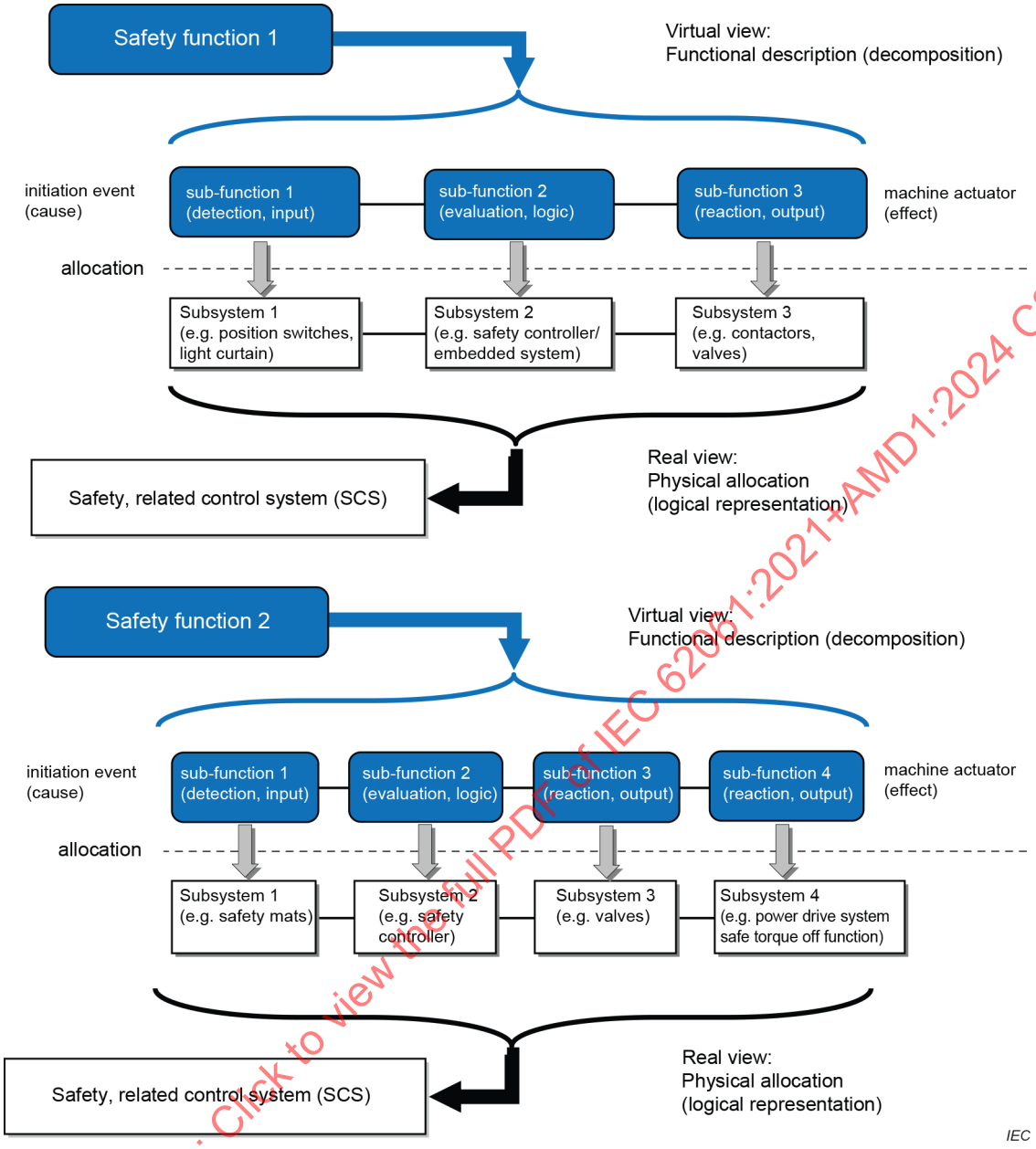
NOTE 3 An SCS can consist of one single subsystem. Example for an SCS implementation with a single subsystem is an “Intelligent” sensor unit (e.g. laser scanner) with integrated output switching device (e.g. relay).

NOTE 4 A subsystem which implements a sub-function can consist of more than one physical unit. An example is a safety controller which has separate input, logic, output (and safety-related fieldbus communication) units. The manufacturer can provide separately the safety-related data for the units.
Another example is a safety relay module which monitors the status of an input device. When the safety relay module does not contain enough output contacts for the specific sub-function then an extension safety module can be added. The manufacturer(s) provides separately the safety-related data for all the modules.

NOTE 5 When decomposing safety requirements into sub-requirements, proper documentation and configuration management processes are conducted for ensuring the maintenance of bi-directional traceability between decomposed requirements.

The decomposition of an SCS into subsystems represented in Figure 6 is typical but the whole SCS can be realized by any number of subsystems.

Figure 6 does not present the possible diagnostic functions that can be required to fulfil the safety requirements.



NOTE 1 The fieldbus communication can be part of one or more subsystems.

NOTE 2 Interconnection (e.g. wiring) aspects can be relevant in subsystem(s) (see 7.3.2.2).

Figure 6 – Examples of typical decomposition of a safety function into sub-functions and its allocation to subsystems

6.3.3 Sub-function allocation

Each sub-function shall be allocated to a subsystem within the architecture of the SCS. More than one sub-function (for example implementing different safety functions), can be allocated to a subsystem.

NOTE An example of a subsystem that implements several sub-functions is a safety controller which acts as a logic solver for guard interlocking function and overspeed protection function.

6.3.4 Use of a pre-designed subsystem

The safety performance of a pre-designed subsystem, according to other standards, shall be in line with Table 4.

Table 4 – Required SIL and *PFH* of pre-designed subsystem

IEC 62061 (IEC 61508)	IEC 62061	IEC 61508 [a]	ISO 13849 [b]	IEC 61496
PFH	**SIL**	**at least ...**	**at least ...**	**at least ...**
$< 10^{-5}$	SIL 1	SIL 1	PL b, c	Type 2
$< 10^{-6}$	SIL 2	SIL 2	PL d	Type 3
$< 10^{-7}$	SIL 3	SIL 3	PL e	Type 4
NOTE A relation between IEC 62061 and IEC 61511 (all parts) or ISO 26262 cannot be assumed within this table.				
[a] This column includes SIL-based standards that fulfil the architectural constraints of IEC 61508, such as IEC 61800-5-2 and IEC 60947-5-3. [b] Does not apply to subsystems using complex components, unless they meet the requirements of IEC 61508 or applicable functional safety products standards. Performance Level b does not correspond to SIL1 in case of a category B (ISO 13849-1) structure.				

6.4 Determination of safety integrity of the SCS

6.4.1 General

The SIL(s) that can be achieved by the SCS shall be considered separately for each safety function and shall be determined from the SIL and the *PFH* of each subsystem, as follows:

– the SIL that is achieved is equal to or less than the lowest SIL of any of the subsystems, and

– the SIL is limited by the sum *PFH* value of all subsystems according to Table 3.

Figure 7 shows an example of an SCS with safety integrity of SIL 2 despite the overall *PFH* value being suitable for a higher SIL.

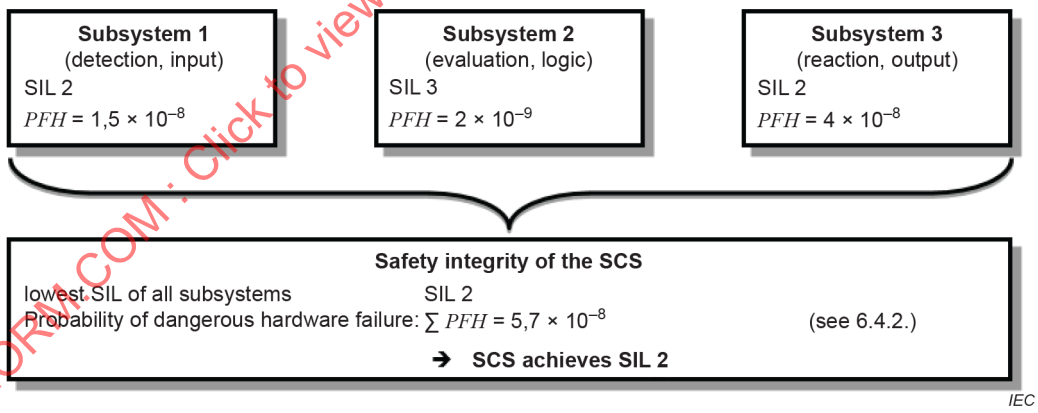


Figure 7 – Example of safety integrity of a safety function based on allocated subsystems as one SCS

NOTE An SCS can be a combination of subsystems based on different architectures.

6.4.2 PFH

The *PFH* of each safety function due to dangerous random hardware failures shall be equal to or lower than the *PFH* of Table 3 related to required SIL as specified in the safety requirements specification.

The estimation of the PFH shall be based on the PFH of each relevant subsystem including, where appropriate, for digital data communication processes between subsystems. The PFH of the SCS is the sum of the probabilities of dangerous random hardware failure of all subsystems involved in the performance of the safety function and shall include, where appropriate, the maximum probability of dangerous transmission errors (P_{TE}) for digital data communication:

$$PFH = PFH_1 + \cdots + PFH_n + P_{TE} \quad (2)$$

NOTE 1 This approach is based on the definition of a subsystem which states that a failure of any subsystem will result in a failure of the SCS (see 6.3.1).

NOTE 2 Hardware wiring aspects are part of systematic integrity and possible failures can be detected by diagnostics.

NOTE 3 For the determination of the P_{TE}, see for example IEC 61784-3.

6.5 Requirements for systematic safety integrity of the SCS

6.5.1 Requirements for the avoidance of systematic hardware failures

The following measures shall apply when appropriate:

a) the SCS shall be designed and implemented in accordance with the functional safety plan (see 4.3);

b) proper selection, combination, arrangements, assembly and installation of subsystems, including cabling, wiring and any interconnections. Wiring interconnection of subsystems may require specific fault considerations and fault exclusions (see 7.3.3);

c) use of the SCS within the manufacturer's specification;

d) use of subsystems that have compatible operating characteristics;

NOTE See also ISO 13849-2:2012, Annexes A, B, C and D.

e) the SCS shall be installed and protected in accordance with IEC 60204-1, including earth fault detection;

f) undocumented modes of component operation shall not be used (e.g. 'reserved' registers of programmable equipment);

g) consideration of foreseeable misuse, environmental changes or modification(s);

h) manufacturer's instructions (including e.g. application examples) of both interconnected subsystems (outputs of the preceding subsystem and inputs of the subsequent subsystem) shall be applied; these can include:

- hardware aspects (e.g. interface information, shielding, signal level, pressure threshold, test pulses, architectural constraints),
- software aspects (e.g. definition of data communication telegrams), and
- diagnostic coverage aspects.

In addition, at least one of the following techniques and/or measures shall be applied taking into account the complexity of the SCS and the SIL(s) for those functions to be implemented by the SCS:

i) SCS hardware design review (e.g. by inspection or walk-through): to reveal by reviews and/or analysis any discrepancies between the specification and implementation;

NOTE 1 In order to reveal discrepancies between the specification and implementation, any points of doubt or potential weak points concerning the realization, the implementation and the use of the product are documented so they can be resolved, taking into account that on an inspection procedure the author is passive and the inspector is active whilst on a walk-through procedure the author is active and the inspector is passive.

j) advisory tools such as computer-aided design packages capable of simulation or analysis, and/or the use of computer-aided design tools to perform the design procedures systematically with the use of pre-designed elements that are already available and tested;

NOTE 2 The integrity of these tools can be demonstrated by specific testing, or by an extensive history of satisfactory use, or by independent verification of their output for the particular SCS that is being designed.

k) simulation: perform a systematic and complete assimilation of an SCS design in terms of both functional performance and the correct dimensioning and interaction of its subsystems.

EXAMPLE The functions of the SCS can be simulated on a computer via a software behavioural model where individual subsystems or subsystem elements each have their own simulated behaviour, and the response of the circuit in which they are connected is examined by looking at the marginal data of each subsystem or subsystem element.

6.5.2 Requirements for the control of systematic faults

The following measures shall be applied:

a) use of de-energization: the SCS shall be designed so that with loss of its supply, a safe state of the machine is achieved or maintained;

b) measures to control the effect of temporary subsystem failures: the SCS shall be designed so that, for example:

- supply variation (e.g. interruptions, dips) to an individual subsystem or a part of a subsystem does not lead to a hazard (e.g. a voltage interruption that affects a motor circuit shall not cause an unexpected start-up when the supply is restored), and

NOTE 1 See also relevant requirements of IEC 60204-1. In particular:

– overvoltage or undervoltage can be detected early enough so that all outputs can be switched to a safe condition by the power-down routine or a switch-over to a second power unit; and/or

– where necessary, overvoltage or undervoltage can be detected early enough so that the internal state can be saved in non-volatile memory, so that all outputs can be set to a safe condition by the power-down routine, or all outputs can be switched to a safe condition by the power-down routine or a switch-over to a second power unit.

See also relevant information in IEC 61131-2.

- the effects of electromagnetic interference from the physical environment or a subsystem(s) do not lead to a hazard;

c) measures to control the effects of errors and other effects arising from any data communication, including transmission errors (such as repetitions, deletion, insertion, re-sequencing, corruption, delay and masquerade);

NOTE 2 Further information can be found in IEC 61784-3:2021, Table 1 and IEC 61508-2:2010, 7.4.11.2.

NOTE 3 The term ‘masquerade’ means that the true contents of a message are not correctly identified. For example, a message from a non-safety component is incorrectly identified as a message from a safety component.

d) when a dangerous fault occurs at an interface, the fault reaction function shall be performed before the hazard due to this fault can occur. When a fault that reduces the hardware fault tolerance to zero occurs, this fault reaction shall take place before the estimated MTTR (see 3.2.39) is exceeded.

The requirements of item d) apply to interfaces that are inputs and outputs of subsystems and all other parts of subsystems that include or require cabling during integration (for example output signal switching devices of a light curtain, output of a guard position sensor).

NOTE 4 This does not require that a subsystem or subsystem element on its own has to detect a fault on its outputs(s). The fault reaction function can also be initiated by any subsequent subsystem after a diagnostic test is performed.

6.6 Electromagnetic immunity

The function of electrical or electronic safety-related systems shall not be affected by external influences in a way that could lead to an unacceptable risk. Acceptable performance with respect to electromagnetic disturbances is therefore mandatory. A comprehensive safety analysis shall include the effects of electromagnetic disturbances and the electromagnetic immunity limits that are required to achieve functional safety. These limits should be derived taking into account both the electromagnetic environment and the required safety integrity levels.

The SCS shall fulfil the applicable requirements of IEC 61000-1-2.

NOTE 1 The appropriate immunity levels in the case of industrial environments are given by IEC 61326-3-1 or IEC 61000-6-7 as a minimum.

NOTE 2 If a subsystem has been designed following an appropriate safety-related product standard (e.g. IEC 61496-1, etc.) or to IEC 61326-3-1 or IEC 61000-6-7, it can be possible that information is supplied with the subsystem that facilitates verification of the SCS level requirements by analysis.

NOTE 3 Guidance design principles are available in EMC standards, but functional safety standards require higher immunity levels. It is important to recognise that higher immunity levels, or additional immunity requirements, than those specified in such standards can be necessary for particular locations or when the equipment is intended for use in harsher, or different, electromagnetic environments.

6.7 Software based manual parameterization

6.7.1 General

Some safety related subsystems or SCS need parameterization to carry out a safety function or a sub-function. For example, a converter with integrated sub-functions has to be parameterized via a PC-based configuration tool, with respect to the upper safe speed limit. Similarly, to properly establish the detection zone of a laser scanner, parameters such as angle and distance can need to be configured per the manufacturer's safety documentation and the machine risk assessment.

The objective of the requirements for software based manual parameterization is to guarantee that the safety-related parameters specified for a safety function or a sub-function are correctly transferred into the hardware performing the safety function or a sub-function. Different methods can be applied to set such parameters; even dip switch based parameterization can be used to set or change safety-related parameters. However, PC-based tools with dedicated parameterization software, commonly called configuration or parameterization tools, are becoming more prevalent. This subclause is limited in scope to only manual, software based parameterization that is performed and controlled by an authorized person.

NOTE 1 Safety-related parameterization which is carried out automatically without human interaction, for example, based on input signals, is not considered in this Subclause 6.7.

NOTE 2 Direct control of a machine by an operator, e.g. speed control of a forklift truck is not considered as manual parameterization as described in this subclause.

NOTE 3 If the configuration or parameterization tool is pre-designed in accordance with IEC 61508-3, for example together with its dedicated subsystem, it is assumed that there will be no dangerous failures due to the influences listed in 6.7.2 or any other influence that is reasonably foreseeable. The requirements of 6.7.5 apply when a software based manual parameterization is performed with the pre-designed tool.

6.7.2 Influences on safety-related parameters

During software based manual parameterization, the parameters can be affected by several influences, such as:

– data entry errors by the person responsible for parameterization;
– faults of the software of the parameterization tool;
– faults of further software and/or service provided with the parameterization tool;

– faults of the hardware of the parameterization tool;
– faults during transmission of parameters from the parametrization tool to the SCS or a subsystem;
– faults of the SCS or a subsystem to store transmitted parameters correctly;
– systematic interference during the parameterization process, e.g. by electromagnetic interference or loss of power;
– interference due to external influences or factors, such as electromagnetic interference or (random) loss of power.

With no measures applied to counteract, avoid or control potential dangerous failures caused by the influences listed above, such influence can lead to the following:

– parameters are not updated by the parameterization process, completely or in parts without notice to the person responsible for the parametrization;
– parameters are incorrect, completely or in parts;
– parameters are applied to an incorrect device, such as when transmission of parameters is carried out via a wired or wireless network.

6.7.3 Requirements for software based manual parameterization

Software based manual parameterization shall use a dedicated tool provided by the manufacturer or supplier of the SCS or the related subsystem(s). This tool shall have its own identification (name, version, etc.). The SCS or the related subsystem(s) and the parameterization tool shall have the capability to prevent unauthorized modification, for example by using a dedicated password.

Parameterization while the machine is running shall be permitted only if it does not cause a hazardous situation.

When using a pre-designed SCS or subsystem that is capable of software based manual parameterization, the objective is to prevent dangerous failure due to the influences listed in 6.7.2 or any other influence that is reasonably foreseeable.

It is possible to fulfil the requirements by using a pre-designed SCS or subsystem, or the design of the SCS or subsystem shall follow this document. Aspects of parametrization shall be included in the validation of the SCS.

The following requirements shall be fulfilled.

a) The design of the software based manual parameterization shall be considered as a safety-related aspect of SCS design that is described in a safety requirements specification, e.g. the software safety requirements specification (see 8.3.2.2 and 8.4.2.2).
b) The SCS or subsystem shall provide means to check the data plausibility, e.g. checks of data limits, format and/or logic input values.
c) The integrity of all data used for parameterization shall be maintained. This shall be achieved by applying measures to
   - control the range of valid inputs;
   - control data corruption before transmission;
   - control the effects of errors from the parameter transmission process;
   - control the effects of incomplete parameter transmission;
   - control the effects of faults and failures of hardware and software of the parameterization; and
   - control the effect of interruption of the power supply.

d) The parameterization tool shall fulfil all relevant requirements for a subsystem according to IEC 61508 to ensure correct parameterization.

e) Alternatively to d) a special procedure shall be used for setting the safety-related parameters. This procedure shall include confirmation of input parameters to the SCS by either:

- retransmitting of modified parameters to the parameterization tool; or
- other means to confirm the integrity of the parameters

as well as subsequent confirmation, for example by a suitably skilled person and by means of an automatic check by a parameterization tool. New values of safety-related parameters shall not be activated before the changes are acknowledged and confirmed.

NOTE This is of particular importance where a parameterization software tool uses a device not specifically intended for this purpose (e.g. personal computer or equivalent).

The software modules used for encoding/decoding within the transmission/retransmission process and software modules used for visualization of the safety-related parameters to the user shall, as a minimum, use diversity in function(s) to avoid systematic failures.

6.7.4 Verification of the parameterization tool

As a minimum, the following verification activities shall be performed to verify the basic functionality of the parameterization tool:

– verification of the correct setting for each safety-related parameter (minimum, maximum and representative values);

– verification that the safety-related parameters are checked for plausibility, for example by detection of invalid values, etc.;

– verification that means are provided to prevent unauthorized modification of safety-related parameters.

NOTE This is of particular importance where the parameterization is carried out using a device not specifically intended for this purpose (e.g. personal computer or equivalent).

6.7.5 Performance of software based manual parameterization

Software based manual parameterization shall be carried out using the dedicated parameterization tool provided by the manufacturer or supplier of the SCS or the related subsystem(s) and shall be documented according to the requirements given in the information for use. This information can originate from different parties, see also 10.3 (information for use). Protective measures against unauthorized access shall be activated and used.

The initial parameterization, and subsequent modifications to the parameterization, shall be documented. The documentation shall include:

a) the date of initial parameterization or change;

b) data or version number of the data set;

c) name of the person carrying out the parameterization;

d) an indication of the origin of the data used (e.g. pre-defined parameter sets);

e) clear identification of safety related parameters;

f) clear identification of the SCS which are subject to specific parametrization settings.

6.8 Security aspects

Security covers intentional attacks on the hardware, application programs and related software, as well as unintended events resulting from human error.

NOTE 1 Security aspects are considered in the security lifecycle of the machine (or higher system level) and throughout the life cycle of the machine.

NOTE 2 Since this document does not provide specific requirements on security aspects, guidance is provided in IEC TS 63074, ISA TR84.00.09, ISO/IEC 27001:2022, ISO TR 22100-4 and IEC 62443 (all parts).

When security countermeasures are applied, they shall not adversely affect safety integrity (e.g. increase in response time, etc.). This can require an iterative multi-disciplinary team analysis.

When security countermeasures implemented within the SCS are declared, then information shall be provided as appropriate.

6.9 Aspects of periodic testing

Periodic testing of the safety function or sub-functions serves two different purposes:

– periodic testing confirms at a given point of time that the tested function is not failed;
– periodic testing in conjunction with inspections assures that the boundary conditions for equipment reliability figures are met.

In general, two types of periodic testing are distinguished:

– diagnostic tests are carried out automatically (initiated automatically or manually) and frequently (related to the process safety time and demand rate);

NOTE 1 Periodic testing can apply to a sub-function or a safety function.

– periodic tests try to verify the complete function, typically by simulating the dangerous condition to the sensors or at least to the logic solver. Also, inspections for ageing and degradation of components are done as part of proof tests.

NOTE 2 The dangerous failures that cannot be detected by the diagnostics are considered to be undetected dangerous failures (related failure rate λ_{DU}). These failures can only be found by the proof-test.

In order to use periodic testing as safety integrity assurance, the following conditions shall be met:

– in the test procedure, a fault reaction shall be implemented to set the relevant parts of the machine in a safe state as consequence of a detected fault;

NOTE 3 The nature of fault reaction can be different for diagnostic and proof test and this also depends on the demand mode and architecture. For architecture of functions with HFT 0 and high or continuous demand, it is usually required to immediately shut-down the machinery.

– the test interval shall be adequate to reveal failures in respect to demand rate;
– for diagnostic tests, see also 7.4.3 for specific requirements.

7 Design and development of a subsystem

7.1 General

The subsystem shall be designed in accordance with its safety requirements specification (see 5.2), including basically:

– the functional requirements;
– the requirements for hardware safety integrity:
  - architectural constraints (see 7.4) and
  - *PFH* (see 7.6);
– the requirements for systematic integrity (see 7.3.2 and estimation of CCF in Annex E);
– the requirements for subsystem behaviour on detection of a fault (fault reaction) (see 7.4.3);
– the requirements for software (see Clause 8).

The following information of Table 5 shall be available where relevant for each subsystem during the design and development.

Table 5 – Relevant information for each subsystem

Functional description	
1)	A functional description of the function(s) and interface(s) of the subsystem
Hardware information	
2)	The estimated rates of failure (due to random hardware failures and failure modes) for each subsystem element which could cause a dangerous failure of the subsystem (see Annex C)
3)	Any test and/or maintenance requirements
4)	The probability of dangerous communication errors for digital data communication processes, where applicable
Environmental conditions	
5)	The environment and operating conditions which should be observed in order to maintain the validity of the estimated rates of failure due to random hardware failures
6)	The useful lifetime (see 7.3.4.2) of the subsystem which should not be exceeded, in order to maintain the validity of the estimated rates of failure due to random hardware failures
Design information	
7)	The diagnostic coverage and/or safe failure fraction and the diagnostic test interval (see 7.4.3 and 7.4.4)
8)	Limits on the application of the subsystem which should be observed in order to avoid or control systematic failures
9)	Information which is required to identify the hardware and software configuration of the subsystem
10)	The highest SIL that can be claimed for a safety function under consideration which uses the subsystem on the basis of: – architectural constraints, – measures and techniques used to avoid or control systematic faults being introduced during the design and implementation of the hardware and software of the subsystem, and – the design features that make the subsystem tolerant against systematic faults. NOTE One subsystem can implement sub-functions of several safety functions with different SIL.

7.2 Subsystem architecture design

The architecture of a subsystem is defined by a process of functional decomposition similar as that of the complete safety function that leads to the SCS architecture – see 6.3.2: The specific sub-function of the subsystem can be decomposed into sub-functions of the next lower order which are then assigned to subsystem elements.

As a result, a set of subsystem element(s) can be defined that meets the functional requirements and the integrity requirements of the sub-function.

NOTE 1 A subsystem can be designed by using one single subsystem element.

NOTE 2 The decomposition into subsystem element(s) can be an iterative process.

NOTE 3 The failure of a subsystem element does not necessarily result in a failure of the subsystem or sub-function. Where subsystem elements are parts of redundant channels, a single element failure will not result in a failure of the safety function.

The design of the subsystem architecture shall be documented in terms of its subsystem elements and their interrelationships, e.g. circuit diagram with description, safety-related block diagram.

Subsystem(s) incorporating complex components shall comply with appropriate product standards or IEC 61508-2 and IEC 61508-3 as appropriate for the required SIL and the design shall use Route 1_H (see IEC 61508-2:2010, 7.4.4.2) for high demand and\or continuous mode.

Where a subsystem design includes such a complex component as a subsystem element, it can be considered as a low complexity component in the context of a subsystem design since its relevant failure modes, behaviour on detection of a fault, rate of failure, and other safety-related information are known. Such components shall only be used in accordance with its specification and the relevant information for use provided by its manufacturer.

NOTE 4 In this document, it is presumed that the design of complex programmable electronic subsystems or subsystem elements conforms to the relevant requirements of IEC 61508 and uses Route 1_H (see IEC 61508-2:2010, 7.4.4.2).

7.3 Requirements for the selection and design of subsystem and subsystem elements

7.3.1 General

There are two types of requirements to subsystems and subsystem elements:

– qualitative requirements: systematic integrity; fault consideration(s) and fault exclusion(s);

– quantitative requirements: failure rate and other relevant parameters.

Qualitative requirements are defined in the following Subclauses 7.3.2 and 7.3.3. Where not explicitly stated otherwise, these requirements apply independently of the SIL requirement to the safety function from SIL 1 up to SIL 3.

NOTE SIL 4 is not considered in this document, as it is not suitable to the risk reduction requirements associated with machinery. For requirements applicable to SIL 4, see IEC 61508-1 and IEC 61508-2.

The quantitative requirements are described in 7.4 in general terms and for determination of the PFH, refer to 6.3.2 and 7.6.

7.3.2 Systematic integrity

7.3.2.1 General

The systematic safety integrity requirements for a subsystem are met by fulfilling the requirements in 7.3.2.2 and 7.3.2.3 and are the same for SIL 1, SIL 2 and SIL 3.

NOTE The subsystem can be partitioned into subsystem elements, pre-designed in agreement with IEC 61508, with different systematic capability level. Then the systematic capability of one subsystem element can potentially limit the SIL of its subsystem. For additional details, see IEC 61508-2.

7.3.2.2 Requirements for the avoidance of systematic failures

The following measures shall all be applied if applicable:

– appropriate selection, combination, arrangements, assembly and installation of components, including cabling, wiring and any interconnections:
apply manufacturer's application notes, e.g. user manual, installation instructions, specifications and use of good engineering practice (e.g. IEC 60204-1);

– use of the subsystem and subsystem elements within the manufacturer’s specification and installation instructions;

– compatibility: use components with compatible operating characteristics;

– withstanding specified environmental conditions:
design the subsystem so that it is capable of working in all expected environments and in any foreseeable adverse conditions (within the defined limit of use), for example temperature, humidity, vibration and electromagnetic fields;

– use of components that are in accordance with an applicable standard and have their failure modes well-defined: to reduce the risk of undetected faults by the use of components with specific characteristics;

NOTE 1 Components such as hydraulic or pneumatic valves can require cyclic switching to avoid the failure mode of non-switching or unacceptable increase in switching times. In this case, a periodic test can be necessary.

– use of suitable materials and adequate manufacturing:
selection of material, manufacturing methods and treatment in relation to, for example stress, durability, elasticity, friction, wear, corrosion, temperature, conductivity, dielectric strength;

– correct dimensioning and shaping:
consider the effects of, for example, stress, strain, fatigue, temperature, surface roughness, manufacturing tolerances.

NOTE 2 IEC 61508-2:2010, Annex F specifies techniques and measures for avoidance of systematic failures during design and development of application-specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), programmable logic devices (PLDs), etc.

NOTE 3 Table B.1 to B.5 of IEC 61508-2:2010, Annex B give techniques and measures to avoid failures in safety-related systems which can be useful during specification, design, integration, operation, maintenance and validation phases.

NOTE 4 Annexes A to D of ISO 13849-2:2012 provide principles for mechanical, pneumatic, hydraulic and electrical systems.

In addition, one or more of the following measures shall be applied if applicable:

a) hardware design review (e.g. by inspection or walk-through):
to reveal by reviews and/or analysis discrepancies between the specification and implementation;

NOTE 5 In order to reveal discrepancies between the specification and implementation, any points of doubt or potential weak points concerning the realization, the implementation and the use of the product are documented so they can be resolved; in an inspection procedure the author is passive and the person inspecting is active whilst on a walk-through procedure the author is active and the person inspecting is passive.

b) computer-aided design tools capable of simulation or analysis:
perform the design procedure systematically and include appropriate automatic construction elements that are already available and tested;

NOTE 6 These tools can be qualified by specific testing, or by an extensive history of satisfactory use, or by independent verification of their output for the particular subsystem that is being designed.

c) simulation:
perform a systematic simulation of a subsystem design in terms of both the functional performance and the correct dimensioning of their components.

NOTE 7 The function of the subsystem can be simulated on a computer via a software behavioral model where individual components of the circuit each have their own simulated behaviour, and the response of the subsystem in which they are connected is examined by looking at the marginal data of each component.

7.3.2.3 Requirements for the control of systematic failures

The following measures shall all be applied if applicable:

a) measures to control the effects of insulation breakdown, voltage variations and interruptions, overvoltage and undervoltage: subsystem behaviour in response to insulation breakdown, voltage variations and interruptions, overvoltage and undervoltage conditions shall be pre-determined so that the subsystem can achieve or maintain a safe state;

NOTE 1 Further information can be found in IEC 60204-1 and IEC 61508-7:2010, Clause A.8.

b) measures to control or avoid the effects of the physical environment (for example, temperature, humidity, water, vibration, dust, corrosive substances, electromagnetic interference and its effects): subsystem behaviour in response to the effects of the physical environment shall be pre-determined so that the SCS can achieve or maintain a safe state. See also e.g. IEC 60529, IEC 60204-1 and IEC 60721 (all parts);

c) measures to control or avoid the effects of temperature increase or decrease, if temperature variations can occur: the subsystem should be designed so that, for example, over-temperature can be detected before it begins to operate outside specification;

NOTE 2 Further information can be found in IEC 61508-7:2010, Clause A.10.

d) measures to control the effects of hose breakdown, pressure variations and interruptions, too low or too high pressure: subsystem behaviour in response to hose breakdown, pressure variations and interruptions, too low or too high pressure shall be pre-determined so that the subsystem can achieve or maintain a safe state.

NOTE 3 Further information can be found in ISO 4414:2010 for pneumatic systems or ISO 4413 for hydraulic systems.

When PELV/SELV power supply (see IEC 60364-4-41) is used, the over voltage at the output in event of a single fault shall be taken into account in the analysis of the effects of over voltage including the possibility of common cause failure.

NOTE 4 Over voltage ranges are given for example in IEC 60950-1, IEC 61204-7, IEC 62477 (all parts), IEC 60449.

In addition, the following basic safety principles, as appropriate, shall be applied for the control of systematic failures:

– use of de-energization:
the subsystem should be designed so that with loss of its power supply, a safe state can be achieved or maintained;

NOTE 5 For further information, see ISO 13849-2.

– measures for controlling the effects of errors and other effects arising from any data communication process (see IEC 61508-2:2010, 7.4.11).

Depending on the selected architecture of the subsystem, the following well tried safety principles, as appropriate, shall be applied to the subsystem element for the control of systematic failures:

– failure detection by automatic tests;
– tests by comparison of redundant hardware;

NOTE 6 For further information, see ISO 12100:2010, 6.2.12.4.

– diverse hardware;
– operation in the positive mode (e.g. a limit switch is pushed when a guard is opened);
– mechanically linked contacts;
– direct opening action;
– oriented mode of failure;

NOTE 7 For further information, see ISO 12100:2010, 6.2.12.3.

– over-dimensioning by a suitable factor can improve reliability and an appropriate factor of over-dimensioning shall be determined.

NOTE 8 For further information, see ISO 13849-2 and Annex A of IEC 61508-2:2010.

7.3.2.4 Electromagnetic immunity

Subsystem design shall take into account the requirements of 6.6.

7.3.2.5 Security aspects

Subsystem design shall take into account the requirements of 6.8.

7.3.3 Fault consideration and fault exclusion

7.3.3.1 General

All subsystem elements shall be designed to achieve the required safety requirement specification. The ability to resist faults shall be assessed. Where not explicitly stated otherwise, the requirements of this Clause 7 apply independently of the required safety integrity of the safety function.

7.3.3.2 Fault consideration

To estimate the capability of a subsystem element to reach a certain safe state, an analysis of each subsystem element shall be performed to determine all relevant faults and their corresponding failure modes. Whether a failure is a safe or a dangerous failure depends on the SCS and the intended safety functions, including fault reaction function.

Analysis technique such as failure mode and effect analysis (FMEA, see IEC 60812), fault tree analysis (FTA, see IEC 61025) or event tree analysis (ETA, see IEC 62502) can be carried out to establish the faults that are to be considered for those components.

The probability of each failure mode shall be determined based on the probability of the associated fault(s) taking into account the intended use and can be derived from sources such as:

– dependable failure rate data collected from field experience by the manufacturer and relevant to the intended use;
– component failure data from a recognised industry source and relevant to the intended use;
– failure mode data;
– failure rate data derived from the results of testing and analysis.

In general, 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 (known as a dependent 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.

7.3.3.3 Fault exclusion

It is not always possible to evaluate subsystems without assuming that certain faults may be excluded. Fault exclusion is a compromise between technical safety requirements and the theoretical possibility of occurrence of a fault.

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.

Fault exclusion is only applicable for certain failures of an element and it is up to the designer (manufacturer or integrator) to prove the exclusion of the respective faults based on the limits set forward by the design and use. Such fault exclusion is only possible provided that the technical improbability of them occurring can be justified based on the known laws of physical science. Any such fault exclusions shall be justified and documented.

The application of fault exclusion to certain faults for an element inside a subsystem does not limit the necessity of the application of systematic measures.

It is possible some faults are excluded by the manufacturer and some by the subsystem integrator.

Fault exclusion is one principle to limit the failure of a component/subsystem; also other methods are possible (e.g. architectures, limitation of systematic failures).

There shall be a specific characterization of the type of fault that is excluded. It would not be acceptable to state simply that a component will not break, distort or degrade due to wear. It would be necessary to state the direct influence under which the component will not break, distort or degrade due to wear. E.g. the component will have no faults when subjected to a force of X Newtons from direction Y.

The fault exclusion shall be justifiable under all expected industrial environments including temperature, pressure, vibration, pollution, corrosive atmosphere, etc.

NOTE 1 Useful information on fault exclusions is available in ISO 13849-2:2012, Annex A to D.

Fault exclusion can only be applied for the entire subsystem when all dangerous failures of a subsystem can be excluded.

LIMITATION: For some applications, it is not expected that all failures can be excluded with sufficient confidence for SIL 3. The following non exhaustive list provides an indication of (non-predesigned) subsystems with a hardware fault tolerance of zero and where fault exclusions have been applied to faults that could lead to a dangerous failure where a maximum of SIL 2 can be appropriate provided that sufficient justification is given:

– position switch with mechanical aspects with HFT of 0;
– leakage of a fluid power valve (where leakage is dangerous failure).

NOTE 2 This limitation does not apply to pre-designed subsystems used within their specification.

7.3.3.4 Functional testing to detect fault accumulation and undetected faults

In a redundant system, an accumulation of faults over time might lead to a loss of the safety function. In a single channel system, undetected faults might also lead to a loss of the safety function.

For an SCS with non-electronic technology and using automatic monitoring to achieve the necessary diagnostic coverage for the required safety performance, the monitoring function cannot be possible unless there is a change of state, e.g. at every operating cycle. If, in such a case, there is only infrequent operation, the probability of occurrence of an undetected fault is increased. When a functional test is necessary to detect a possible accumulation of faults or a undetected fault before the next demand, it shall be made within the following test intervals:

– at least every month for SIL 3;
– at least every 12 months for SIL 2.

EXAMPLE: The control system of a machine can demand these tests at the required intervals e.g. by visual display unit or signal lamp, and can monitor the tests and stop the machine if the test is omitted or fails.

7.3.4 Failure rate of subsystem element

7.3.4.1 General

The mathematical probability of failure of a subsystem element can be characterized by one of three parameters: λ (Lambda), $MTTF$ (Mean Time To Failure) or B_{10}.

NOTE Although the parameters above can be delivered in several valuable formats, the typical formats are:

– λ: failures per hour;

– $MTTF$: mean time to failures expressed in years;

– B_{10}: switching cycles of wearing components.

For the estimation of the parameters of a subsystem element, the hierarchical procedure for finding data shall be, in the order given:

a) use manufacturer's data;

b) use Annex C of this document;

c) choose a $MTTF_D$ of ten years.

The data could be delivered as values with respect to the dangerous failures (λ_D, $MTTF_D$, B_{10D}) or with respect to all failures (λ, $MTTF$, B_{10}).

To determine the dangerous failures from the overall failures, the different failure modes of the subsystem element should be taken into account. It is typically assumed that not all failures modes lead to a dangerous failure. This depends mainly on the application, so generally the failure mode data used should reflect practical application of the components. A precise way of determining the "failure modes" of a subsystem element is to carry out an FMEA. If no specific or sufficient knowledge and information is available concerning the failure modes, 50 % of the failures can be estimated as dangerous.

7.3.4.2 Relationship of relevant parameters

For subsystem elements, constant failure rates (λ) of the subsystem elements are assumed. The following basic equations can be used:

$$\lambda = \frac{1}{MTTF} \tag{3}$$

$$\lambda_D = \frac{1}{MTTF_D} \tag{4}$$

NOTE 1 For calculation purposes, $MTTF$ can be assumed equal to mean operating time between failures ($MTBF$).

$MTTF$ and $MTTF_D$ are mostly indicated in years [a]. λ values are commonly indicated in FIT (FIT = Failure In Time) where 1 FIT means one failure in 10^9 hours.

$$1\,FIT = 1 \times 10^{-9}\,\mathrm{h}^{-1} \tag{5}$$

One year is approximately 8 760 hours. Therefore, a $MTTF$ value can be converted into a λ value.

$$\lambda = \frac{1}{MTTF \times 8760\,\frac{\mathrm{h}}{\mathrm{a}}} \tag{6}$$

NOTE 2 Example, $MTTF$ = 1 000 a:

$$\lambda_{example} = \frac{1}{1\,000a \times 8760\frac{h}{a}}$$

$$\lambda_{example} = \frac{1}{8\,760\,000h}$$

$$\lambda_{example} = \frac{1}{8\,760\,000}h^{-1}$$

$$\lambda_{example} = 114{,}155 \times 10^{-9}h^{-1}$$

$$\lambda_{example} = 114{,}155\ FIT$$

For pneumatic, mechanical and electromechanical components (pneumatic valves, relays, contactors, position switches, cams of position switches, etc.) it can be difficult to calculate the mean time to dangerous failure ($MTTF_D$ for components), which is given in years. Usually the manufacturers of these kinds of components only give the mean number of cycles until 10 % of the components fail dangerously (B_{10D}). This Clause 7 gives a method for calculating an $MTTF_D$ for components by using B_{10D} given by the manufacturer related closely to the application dependent cycles.

NOTE 3 Hydraulic components are mostly characterized with $MTTF_D$.

If the appropriate basic and well-tried safety principles are met, the $MTTF_D$ value for a single pneumatic, electromechanical or mechanical component can be estimated.

The mean number of cycles until 10 % of the components fail dangerously (B_{10D}) should be determined by the manufacturer of the component in accordance with relevant product standards for the test methods (e.g. IEC 60947-5-1, ISO 19973, IEC 61810). The dangerous failure modes of the component have to be defined, e.g. sticking at an end position or change of switching times. If not all the components fail dangerously during the tests (e.g. seven components tested, only five fail dangerously), an analysis taking into account the components that were not dangerously failed components should be performed.

With B_{10D} and n_{op}, the mean number of annual operations, $MTTF_D$ for components can be calculated as

$$MTTF_D = \frac{B_{10D}}{0{,}1\ \ n_{op}} \tag{7}$$

where

$$n_{op} = \frac{d_{op} \times h_{op} \times 3600\frac{s}{h}}{t_{cycle}} \tag{8}$$

and with the following assumptions having been made on the application of the component:

h_{op} is the mean operation, in hours per day;

d_{op} is the mean operation, in days per year;

t_{cycle} is the mean time between the beginning of two successive cycles of the component. (e.g. switching of a valve) in seconds per cycle.

In terms of failure rate λ, the following relationship can be expressed as

$$\lambda_D = \frac{0{,}1\ C}{B_{10D}} = \frac{0{,}1\ n_{op}}{B_{10D} \times 8\ 760 \frac{h}{a}} \tag{9}$$

where C (C = n_{op} / 8 760) is the duty cycle or mean operation per hour.

The relation between B_{10D}, B_{10} and the ratio of dangerous failure (RDF) is

$$B_{10D} = \frac{B_{10}}{\text{ratio of dangerous failure}} \tag{10}$$

The useful lifetime of the component is limited to T_{10D}, the mean time until 10 % of the components fail dangerously:

$$T_{10D} = \frac{B_{10D}}{n_{op}} \tag{11}$$

NOTE 4 For electronic systems, the exponential distribution is applicable. For non-electronic systems, the exponential distribution is not applicable. The Weibull distribution (see also IEC 61649) is more appropriate, but parameters and calculations are difficult to apply. However, when using exponential distribution for non-electronic components within the limits of T_{10D} then the results of the calculations are pessimistic and the formula with $1\text{-}e^{-\lambda t}$ could be applied as a simplified method.

If the ratio of dangerous failure is estimated less than 0,5 (50 % dangerous failure) the useful lifetime of the component is limited to twice T_{10}.

NOTE 5 Similar to Formula (11), T_{10} is evaluated by $T_{10} = \frac{B_{10}}{n_{op}}$.

For further details, see IEC TS 63394:2023, Clause H.6.

The ratio of dangerous failure is estimated as 0,5 (50 % dangerous failure) if no other information (e.g. product standard) is available.

7.4 Architectural constraints of a subsystem

7.4.1 General

In the context of hardware safety integrity, the highest safety integrity level that can be claimed for an SCS is limited by the hardware fault tolerances (HFT) and safe failure fractions (SFF) of the subsystems that carry out that safety function. Table 6 specifies the highest safety integrity level that can be claimed for an SCS that uses a subsystem taking into account the hardware fault tolerance and safe failure fraction of that subsystem. The architectural constraints given in Table 6 shall be applied to each subsystem developed according to Clause 7. With respect to these architectural constraints:

a) a hardware fault tolerance of N means that N+1 faults could cause a loss of the safety function. In determining the hardware fault tolerance, no account is taken of other measures that can control the effects of faults such as diagnostics; and

b) where one fault directly leads to the occurrence of one or more subsequent faults, these shall be considered as a single fault;

c) in determining hardware fault tolerance, certain faults may be excluded, provided that the likelihood of them occurring is very low in relation to the safety integrity requirements of the subsystem, see 7.3.3.3.

A subsystem that comprises only a single subsystem element shall satisfy the requirements of Table 4. In particular, for an HFT 0 (zero fault tolerance) subsystem element of SIL 3, a SFF of greater than 99 % shall be achieved by an SCS diagnostic function.

When two or more pre-designed subsystems are combined into one redundant subsystem, the architectural constraints of the combined subsystem can be determined. This can be done by taking the subsystem with the highest SIL according to the architectural constraints and looking for the corresponding SIL in Table 6 in column HFT 0. This will return the applicable SFF range. The SIL of the combined subsystem shall be derived by increasing the HFT by one in the same SFF range according to IEC 61508-2:2010, 7.4.4.2.4

NOTE 2 This procedure is only applicable for combining subsystems with a defined SIL.

Table 6 – Architectural constraints on a subsystem: maximum SIL that can be claimed for an SCS using the subsystem

Safe failure fraction (SFF)	Hardware fault tolerance (HFT) (see NOTE 1)		
	0	1	2
< 60 %	Not allowed (for exceptions see NOTE 3)	SIL 1	SIL 2
60 % to < 90 %	SIL 1	SIL 2	SIL 3
90 % to < 99 %	SIL 2	SIL 3	SIL 3 (see NOTE 2)
≥ 99 %	SIL 3	SIL 3 (see NOTE 2)	SIL 3 (see NOTE 2)

NOTE 1 A hardware fault tolerance of N means that N+1 faults could cause a loss of the safety function.

NOTE 2 SIL 4 is not considered in this document. For SIL 4, see IEC 61508-1.

NOTE 3 See 7.5.3, where subsystems which have a safe failure fraction of less than 60 % and zero hardware fault tolerance that use well-tried components can be considered to achieve SIL 1; or for subsystems where fault exclusions have been applied to faults that could lead to a dangerous failure.

NOTE 4 In IEC 62061:2015 the maximum SIL that could be claimed was named SILCL.

NOTE 5 See 7.3.3.3 for limitation of SIL when applying fault exclusion.

NOTE 6 For HFT 0 at $SFF \geq 99$ %, it is only possible when there is continuous monitoring of the correct functioning of the element. Typically, electronic technology will be required to achieve this.

7.4.2 Estimation of safe failure fraction (SFF)

To estimate the SFF, an analysis (e.g. fault tree analysis, failure mode and effects analysis) of each subsystem shall be performed to determine all relevant faults and their corresponding failure modes. Whether a failure is a safe or a dangerous failure depends on the SCS and the intended safety function, including fault reaction function (see 7.4.3). The probability of each failure mode shall be determined based on the probability of the associated fault(s) taking into account the intended use and may be derived from sources such as:

a) dependable failure rate data collected from field experience by the manufacturer and relevant to the intended use;

b) failure rate data from a recognised industry source and relevant to the intended use;

c) failure rate data derived from the results of testing and analysis.

NOTE 1 Information of the failure rates for electrical/electronic component can be found in several sources including: MIL-HDBK 217 F, MIL-HDBK 217 F (Appendix A), SN 29500 Parts 7 and 11, IEC 61709, FMD-2016, OREDA Handbook, EXIDA Safety Equipment Reliability Handbook and EXIDA Electrical & Mechanical Component Reliability Handbook.

NOTE 2 Failure rate data can be provided by manufacturers.

NOTE 3 Some component standards provide relevant data (e.g. Annex K of IEC 60947-4-1:2018).

NOTE 4 Lists of faults to be considered for mechanical, pneumatic, hydraulic and electrical technologies are given in Annexes A, B, C and D of ISO 13849-2:2012.

In general, the SFF can be calculated as follows:

$$SFF = \frac{\sum \lambda_S + \sum \lambda_{DD}}{\sum \lambda_S + \sum \lambda_D} \tag{12}$$

where

λ_S is the rate of safe failure,

$\sum \lambda_S + \sum \lambda_D$ is the overall failure rate,

λ_{DD} is the rate of dangerous failure which is detected by the diagnostic functions,

λ_D is the rate of dangerous failure.

The failure of an element that plays a part in implementing the safety function but has no direct (adverse) effect on the safety function is termed a no effect failure and is not considered as a safe failure (λ_S). Therefore, it shall not be used for SFF calculations.

For non-electronic components, λ_S is typically assumed as equal to 0 or is negligible, because in most cases it is insignificant in comparison to λ_D. In this case, the following simplification can be applied (see also example in Clause B.4):

$$SFF = \frac{\sum \lambda_S + \sum \lambda_{DD}}{\sum \lambda_S + \sum \lambda_D} \approx \frac{\sum \lambda_{DD}}{\sum \lambda_D} \tag{13}$$

EXAMPLE 1 Where the hardware fault tolerance of a subsystem is equal to 0, the SFF becomes

$$SFF \approx \frac{\lambda_{DD1}}{\lambda_{D1}} = \frac{DC_1 \lambda_{D1}}{\lambda_{D1}} = DC_1$$

where DC_1 is the diagnostic coverage of subsystem element 1.

EXAMPLE 2 Where the hardware fault tolerance of a subsystem is equal to 1, SFF becomes

$$SFF \approx \frac{\lambda_{DD1} + \lambda_{DD2}}{\lambda_{D1} + \lambda_{D2}} = \frac{DC_1 \lambda_{D1} + DC_2 \lambda_{D2}}{\lambda_{D1} + \lambda_{D2}} = \frac{\frac{DC_1}{MTTF_{D1}} + \frac{DC_2}{MTTF_{D2}}}{\frac{1}{MTTF_{D1}} + \frac{1}{MTTF_{D2}}}$$

where DC_1 and DC_2 are the diagnostic coverages respectively of subsystem element 1 and 2 (see also 7.4.2 for relationship between λ and $MTTF$).

7.4.3 Behaviour (of the SCS) on detection of a fault in a subsystem

7.4.3.1 General

The detection of a dangerous fault in any subsystem that has a hardware fault tolerance of more than zero shall result in the performance of the specified fault reaction function.

The specification can allow isolation of the faulty part of the subsystem to continue safe operation of the machine while the faulty part is repaired. In this case, if the faulty part is not repaired within the estimated maximum time, as assumed in the calculation of the PFH, then a second fault reaction shall be performed to achieve a safe state.

Where the SCS is designed for online repair, isolation of a faulty part shall only be applicable where this does not increase the PFH of the SCS above that specified in the SRS.

As long as operation is continued and hardware fault tolerance is reduced to zero, the requirements of 7.4.3.2 apply.

7.4.3.2 Fault reaction function

Where a diagnostic function is necessary to achieve the required PFH or safe failure fraction and the subsystem has a hardware fault tolerance of zero, then

– the sum of the diagnostic test interval and the time to perform the specified fault reaction function to achieve or maintain a safe state shall be shorter than the process safety time (e.g. see ISO 13855); or,

– when operating in high demand mode of operation, the ratio of the diagnostic test rate to the demand rate shall equal or exceed 100.

Where performance of a fault reaction function as part of an SCS that is specified as SIL 3 has resulted in the machine being stopped, subsequent normal operation of the machine via the SCS (e.g. enabling re-start of the machine) shall not be possible until the fault has been repaired or rectified. For an SCS with a specified safety performance of less than SIL 3, the behavior of the machine after performance of a fault reaction function (e.g. re-starting normal operation) shall depend on the specification of relevant fault reaction functions (see 5.2.2).

7.4.3.3 Diagnostic coverage (DC)

Diagnostic coverage (DC) can be calculated as the fraction of dangerous failures by using the following equation:

$$DC = \frac{\sum \lambda_{DD}}{\sum \lambda_{D}} \tag{14}$$

where λ_{DD} is the rate of detected dangerous hardware failures and λ_{D} is the rate of dangerous hardware failures.

For the estimation of DC, in most cases, failure mode and effects analysis (FMEA – see IEC 60812), failure mode effects and diagnostic analysis (FMEDA) or equivalent methods can be used. In this case, all relevant faults and/or failure modes should be considered.

For a simplified approach to estimating DC, see Annex D.

NOTE Annex C of IEC 61508-2:2010 provides further information.

7.4.4 Realization of diagnostic functions

Each subsystem shall be provided with associated diagnostic functions that are necessary to fulfil the requirements for architectural constraints and the PFH.

The diagnostic functions are considered as separate functions that can have a different structure than the safety function and can be performed by

– the same subsystem which requires diagnostics; or
– other subsystems of the SCS; or
– subsystems of the SCS not performing the safety function.

Diagnostic functions shall satisfy the following:

– applicable requirements for the avoidance of systematic failure; and
– applicable requirements for the control of systematic failure.

NOTE 1 Timing constraints applicable to the testing of the subsystem performing a diagnostic function can differ from those applicable to safety functions.

NOTE 2 The necessity of checking the diagnostic function can depend on aspects such as the safety integrity level, the demand rate, the technology used and application specific capabilities.

A clear description of the SCS diagnostic function(s), their failure detection/reaction, and an analysis of their contribution towards the safety integrity of the associated safety functions shall be provided.

To apply the simplified approach of this document for the estimation of PFH of subsystems, the following shall apply:

SCS diagnostic function(s) shall as a minimum be implemented so that the PFH and the systematic safety integrity are the same as those specified for the corresponding safety function(s),

or

where the PFH is of an order of magnitude greater than that specified for the safety function, then a test shall be performed to determine whether diagnostic function(s) remain operational; a test of the diagnostic function(s) shall be carried out at a minimum of 10 times at equal intervals during the proof test interval for the subsystem.

NOTE 3 Architectural constraints on hardware safety integrity do not apply to the realization of diagnostic function(s).

NOTE 4 A test of the diagnostic function(s) is foreseen to cover, as far as practicable, 100 % of those parts implementing the diagnostic function(s).

NOTE 5 Where a diagnostic function is implemented by the logic solver of the SCS, it can be unnecessary to perform a separate test of the diagnostic function as its failure can be revealed as a failure of the safety function.

NOTE 6 A test can be performed by either external means (e.g. test equipment) or internal dynamic checks (e.g., embedded within the logic solver) of the SCS.

7.5 Subsystem design architectures

7.5.1 General

The architecture of any subsystem described in this subclause can be used to evaluate the architectural constraints and to estimate the PFH; see Annex H.

NOTE The figures in 7.5 represent a logical view of the subsystem architectures and are not intended to represent any specific physical connection schemes. A hardware fault tolerance of 1 is represented by parallel subsystem elements but the physical connections will depend on the application of the subsystem.

7.5.2 Basic subsystem architectures

7.5.2.1 Basic subsystem architecture A: single channel without a diagnostic function

In this architecture (see Figure 8), any dangerous failure of a subsystem element causes a failure of the safety function. This architecture corresponds to a hardware fault tolerance of 0.

In high or continuous mode of operation, architecture A shall not rely on a proof test.

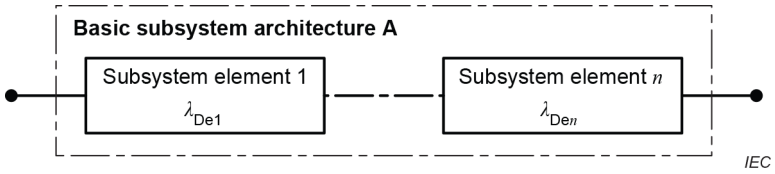


Figure 8 – Basic subsystem architecture A logical representation

7.5.2.2 Basic subsystem architecture B: dual channel without a diagnostic function

This architecture (Figure 9) is such that a single failure of any subsystem element does not cause a loss of the safety function. This architecture corresponds to a hardware fault tolerance of 1.

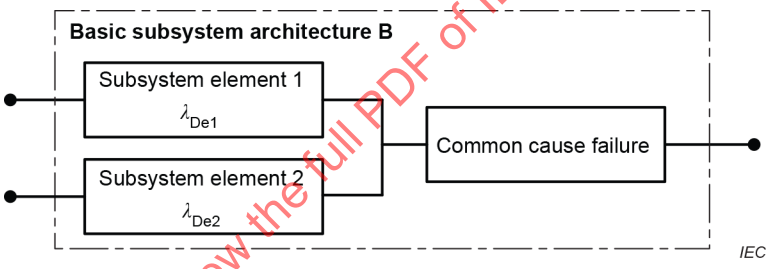


Figure 9 – Basic subsystem architecture B logical representation

7.5.2.3 Basic subsystem architecture C: single channel with a diagnostic function

In this architecture (see Figure 10), any undetected dangerous fault of the subsystem element leads to a dangerous failure of the safety function.

Where a fault of a subsystem element is detected, the diagnostic function(s) initiates a fault reaction function (see 7.4.3). This architecture corresponds to a hardware fault tolerance of 0.

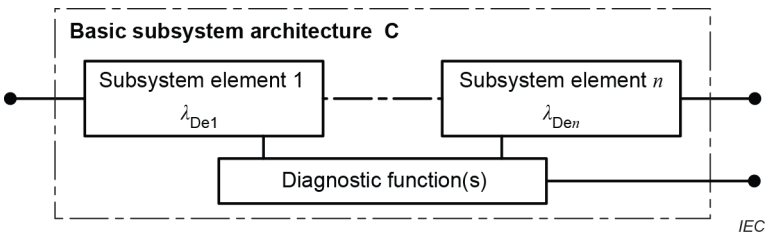


Figure 10 – Basic subsystem architecture C logical representation

7.5.2.4 Basic subsystem architecture D: dual channel with a diagnostic function(s)

This architecture (see Figure 11) is such that a single failure of any subsystem element does not cause a loss of the safety function. Where a fault of a subsystem element is detected, the diagnostic function(s) initiates a fault reaction function (see 7.4.3). This architecture corresponds to a hardware fault tolerance of 1.

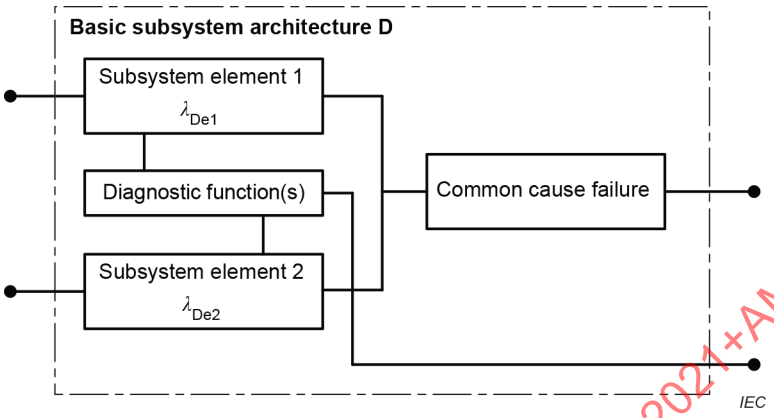


Figure 11 – Basic subsystem architecture D logical representation

7.5.3 Basic requirements

As shown in Table 7, the basic requirements depending on the architectural constraints and the basic subsystem architectures shall be applied.

Table 7 – Overview of basic requirements and interrelation to basic subsystem architectures

Basic requirements	Hardware fault tolerance (HFT)				Comments / Examples
	0		1		
	SFF		*SFF*		
	<60 %	≥ 60 %	<60 %	≥ 60 %	
Basic safety principles	M	M	M	M	Use of suitable materials ISO 13849-2:2012, Annex A to D
Well-tried safety principles	M	M	M	M	Mechanically linked contacts and contacts with direct opening action ISO 13849-2:2012, Annex A to D
Well-tried components	M	--	--	--	Contactor (IEC 60947-4-1) ISO 13849-2:2012, Annex A to D
CCF	not relevant	M	M	M	
Type of basic subsystem architecture	**A**	**C**	**B**	**D**	
M = mandatory; -- = no requirement NOTE Table 6 for architectural constraints is still applicable.					

7.6 *PFH* of subsystems

7.6.1 General

The following parameters have to be determined to be able to determine the PFH:

– subsystem architecture (see 7.5);
– DC and test intervals (see 7.4.3 and 7.4.4);
– CCF (see Annex E);
– λ_D or $MTTF_D$ of subsystem elements (see 7.3.4);
– useful lifetime.

NOTE Because a typical machine life is about 20 years, a useful lifetime of 20 years is preferred. The intention is to clarify the maximum usage period for the subsystem. For components with wear out characteristics, useful lifetime can be limited by T_{10D}.

7.6.2 Methods to estimate the *PFH* of a subsystem

One of the following methods of Annex H may be used to calculate the PFH as simplified approach:

– allocation table approach (see Clause H.1);
– formulas (see Clause H.2).

Modelling based on e.g. fault tree analysis (see B.6.6.5 of IEC 61508-7:2010 and IEC 61025), Markov models (see B.6.6.6, C.6.4 of IEC 61508-7:2010 and IEC 61165) or reliability block diagrams (see C.6.4 of IEC 61508-7:2010) is always possible.

7.6.3 Simplified approach to estimation of contribution of common cause failure (CCF)

Knowledge of the susceptibility of a subsystem to CCF is required to contribute to the estimation of the PFH of a subsystem.

The probability of occurrence of the CCF will usually be dependent upon a combination of technology, architecture, application and environment. The use of Annex E will be effective in avoiding many types of CCF.

8 Software

8.1 General

All lifecycle activities of safety-related application software shall focus on the avoidance of faults introduced during the software lifecycle. The main objective of the following requirements is to produce readable, understandable, testable, maintainable and correct software.

Where the software performs both non-safety and safety functions, then all of the software shall be treated as safety-related, unless sufficient independence between the functions can be demonstrated in the design. It is therefore preferable to separate non-safety functions such as basic machine functions from safety functions wherever practicable.

This document shall only be used for application software that is running in a pre-designed platform according to IEC 61508 or other functional safety standards linked to IEC 61508 e.g. IEC 61131-6.

NOTE In the remainder of this clause, application software is also referred to as software.

8.2 Definition of software levels

This document describes three different levels of application software, see Table 8.

Table 8 – Different levels of application software

SW level	Main principle	Subprinciple	Example
1	Platform (combination of hardware and software) pre-designed according to IEC 61508, or other functional safety standards linked to IEC 61508 e.g. IEC 61131-6. Application software making use of a limited variability language (LVL).	Application software complying with this document.	Safety PLC with LVL or Safety programmable relay
2	Platform (combination of hardware and software) pre-designed according to IEC 61508, or other functional safety standards linked to IEC 61508 e.g. IEC 61131-6. Application software making use of another language than a limited variability language (LVL).	Application software complying with this document.	Safety PLC with FVL (FVL complying with this document.)
3		Application software complying with IEC 61508-3.	Safety PLC with LVL or FVL (FVL according IEC 61508)

NOTE 1 Software Level 2 is introduced to support Full Variability Language, but limited to SIL 2. For SIL 3 compliant application SW, so-called software level 3, follow IEC 61508-3.

NOTE 2 For other types of platforms no requirements are set forward in this document. IEC 61508-2 and 61508-3 describe how to handle such systems.

The programming language (instruction set) to be used for the application software has to be in the safety-related scope of the platform, pre-designed according to IEC 61508, or other functional safety standards linked to IEC 61508 e.g. IEC 61131-6.

The programming language to be used and the tools (of the software development lifecycle) shall be suitable for the creation of safety related application software on the platform; see 8.4.1.3.

In this context, the platform described in Table 8 shall require only the application software to execute its safety related functionality.

NOTE For example, elements such as systems on chip or microcontroller boards are not platforms in this sense.

Software level 3 is not further described in this document since it is covered by correct application of the respective parts of IEC 61508. A high level of competence is required in order to design according to SW level 2 or 3. Factors that make the use of IEC 61508-3 (software level 3) more appropriate than the use of this document (software level 2) are:

– high degree of complexity of the safety function(s);
– large number of safety functions;
– large project size.

The software safety lifecycle requirements for the different SW levels are detailed in the following subclauses:

– SW level 1: see 8.3;
– SW level 2: see 8.4.

8.3 Software – Level 1

8.3.1 Software safety lifecycle – SW level 1

8.3.1.1 Maximum achievable SIL – SW level 1

The maximum achievable SIL for SW level 1 is SIL 3.

8.3.1.2 Software safety lifecycle model – SW level 1

A software safety lifecycle model which is resolved into distinct phases shall be used (e.g. V-model), including management and documentation activities to achieve the required level of safety.

Any software lifecycle model may be used provided all the objectives and requirements of this Subclause 8.3 are met. Safety-related software shall be validated as described in 9.5.4.

SW level 1 is of reduced complexity due to the use of pre-designed safety-related hardware and software modules. Therefore, the simplified V-model in Figure 12 is applicable. The design of customized, or self-created software modules can be necessary (e.g. in the case of the library modules provided by the component manufacturer being inadequate or not suitable). The design of software modules customized by the designer is an additional activity which shall be carried out according to the V-model in Figure 13.

NOTE 1 A software module (or briefly module) is a functional unit of the software, which is typically only accessible through its input and output interface. It is reusable and facilitates the modular software development. Software modules are often part of a library. In PLC-programming, software modules are functions or function blocks.

NOTE 2 The V-model is a static model used to structure the software design into small parts. It does not introduce any sequence of creation of specifications or implementation. The left side represents requirements; i.e. things to achieve. The right side details testing of the software.

NOTE 3 On the left side of the V-model, the output of each phase is reviewed. Review means to check the output of a phase in the V-model against the requirements of the input of the same phase. The arrow 'Review' represents the first step of the software verification. Further information on the level of independence of review and testing/verification is available in Annex J.

NOTE 4 The lifecycle is accompanied by project management techniques and processes appropriate for the size and scope of the project.

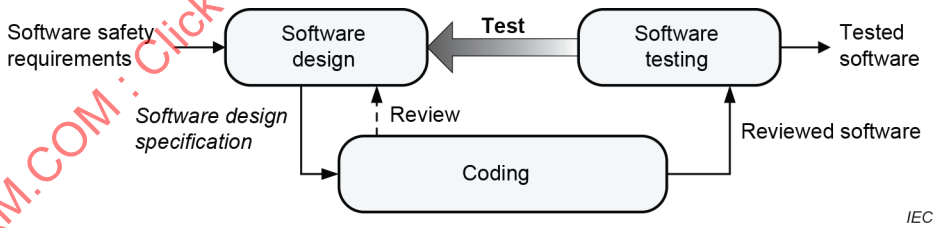


Figure 12 – V-model for SW level 1

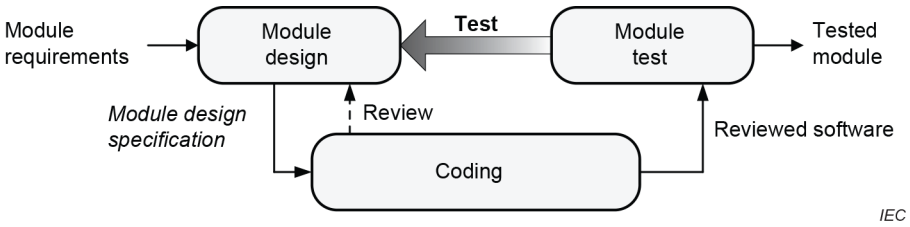


Figure 13 – V-model for software modules customized by the designer for SW level 1

NOTE 5 In the V-models the arrow 'Test' represents the results of test cases according to the specification and, in addition, the need for more precise test case requirements and specifications.

NOTE 6 The result of Figure 13 is an input to the coding of Figure 12.

8.3.1.3 Tools usage – SW level 1

The tools shall be applied according to the instructions of the relevant manufacturer of the safety-related system(s) (e.g. PLC, electro sensitive protective equipment).

8.3.2 Software design – SW level 1

8.3.2.1 General – SW level 1

Where software is to be used in any part of an SCS implementing a Safety Function, a software safety requirements specification shall be developed, documented and managed throughout the lifecycle of the SCS.

The software safety requirements specification shall be developed for each subsystem on the basis of the SCS specification and architecture.

8.3.2.2 Software safety requirements – SW level 1

To support the software design process the following information shall be considered:

a) specification of the safety function(s) (see 5.2);
b) configuration or architecture of the SCS (e.g. hardware architecture, wiring diagram, safety-related inputs and outputs);
c) response time requirements;
d) operator interfaces and controls, such as: switches, joysticks, mode selector, dials, touch sensitive control devices, keypads, etc.;
e) relevant modes of operation of the machine;
f) requirements on diagnostics for hardware including the characteristics of sensors, final actuators, etc.;
g) effects of mechanical tolerances, e.g. of sensors and/or their sensing counter parts;
h) coding guidelines.

8.3.2.3 Software design specification – SW level 1

The software design specification shall be derived from the software safety requirements of the SCS.

The software design specification shall be:

– structured, reviewable, testable, understandable, maintainable and operable;
– developed for each subsystem on the basis of the SCS specification and architecture;
– sufficiently detailed to allow the design and implementation of the SCS to achieve the required level of safety (SIL), and to allow verification and testing;
– traceable back to the specification of the software safety requirements of the SCS. This means that the specification is as such understandable such that another person (e.g. non-software specialist) can verify if the specification corresponds to the software safety requirements of the safety functions defined in the risk assessment;
– free of ambiguous terminology and irrelevant descriptions.

It shall be possible to relate the inputs of the software design specification in a straightforward manner to the desired outputs and vice versa. Where appropriate, easy readable semi-formal methods such as cause&effect tables, logic tables or diagrams, function-blocks or sequence diagrams shall be used in the documentation.

NOTE 1 Where appropriate depends on the number of safety functions involved in the program. Whenever the total amount of safety functions inside the program is larger than 3, it is considered appropriate.

The following shall be specified within the software design specification:

a) logic of the safety functions, including safety-related inputs and outputs and proper diagnostics on detected faults. Possible methods include, but are not limited to, cause&effect table, written description or function blocks;

NOTE 2 Faults can also be detected by hardware (e.g. signal discrepancy detected by input card).

b) test cases that include:
   - the specific input value(s) for which the test is carried out and the expected test results including pass/fail criteria;
   - fault insertion or injection(s);

NOTE 3 For simple functions, the test case(s) can be given implicitly by the specification of the safety function.

c) diagnostic functions for input devices, such as sensing elements and switches, and final control elements, such as solenoids, relays, or contactors;
d) functions that enable the machine to achieve or maintain a safe state;
e) functions related to the detection, annunciation and handling of faults;
f) functions related to the periodic testing of SCS(s) on-line and off-line;
g) functions that prevent unauthorized modification of the SCS (e.g. password);
h) interfaces to non SCS;
i) safety function response time.

NOTE 4 Guidance on software documentation is given in IEC 61508, ISO/IEC/IEEE 26512.

The software design specification shall also explain the main software aspects. Main aspects include for example:

- if appropriate, the software architecture that defines the structure decided to satisfy the software design specification;
- the global data;
- data libraries used;
- pre-existing software modules used;
- diagnostic functions (internal, external);
- programming tools including information which uniquely identifies the tool;
- integration test cases and procedures, including specification of the test environment, support software, configuration description and procedures for corrective action on failure of test.

It is recommended to use pre-designed software modules within the software design specification wherever possible, for example a software module used for muting function according to IEC 61496-1 and designed by the manufacturer of the platform.

It is recommended that in case of pre-designed safety sub-functions, for example IEC 61800-5-2, a reference to the specification provided by the manufacturer should be used.

The information in the software design specification shall be reviewed and where necessary revised, to ensure that the software safety requirements (see 8.3.2.2) are adequately specified.

8.3.3 Module design – SW level 1

8.3.3.1 General – SW level 1

Where previously developed software library modules are to be used as part of the design, their suitability in satisfying the specification of requirements of the software safety shall be demonstrated. Constraints from the previous software development environment (for example operating system and compiler dependencies) shall be evaluated.

8.3.3.2 Input information – SW level 1

For software modules, the following information shall be available in the module requirements:

a) module description;
b) module interface (inputs and outputs with data types, and, if necessary, with data ranges);
c) module libraries used;
d) specific coding rules.

8.3.3.3 Module design specification – SW level 1

The module design specification shall contain the following information:

a) description of the logic (i.e. the functionality) of each module;
b) fully defined input and output interfaces assigned for each module;
c) format and value ranges of input and output data and their relation to modules;
d) test cases which shall include normal and outside normal operation.

NOTE Although test cases usually comprise the individually testing of parameters within their specified ranges, a varying combination of these parameters can introduce unpredicted operation.

This information shall be reviewed against the input information (see 8.3.3.2).

8.3.4 Coding – SW level 1

Software shall be developed in accordance with the design specifications and coding rules. Coding rules can be either well-known industry standards or can be internal to the manufacturer. The code shall be reviewed against the design specifications and coding rules.

NOTE Coding rules are intended to restrict the freedom of programming in order to avoid the program code becoming incomprehensible and in order to reduce the likelihood of the program entering unintended states.

The output of coding shall comprise

– source code listing (e.g. ladder, function blocks, models);
– code review report.

Some typical coding rules to be applied, include, but are not limited to:

– Structure of the program is as easy and clear as possible.
– Structure of the program should be such that the logical flow starts at the top and follows in the effective sequence.
– Every part should have sufficient comments in a predefined way.
– Same names for parameters as during design should be used.
– Names should represent the function of the parameter in a clear way.
– A predefined state should exist.
– Use of set/reset for safety functions should be limited.
– Safety outputs should only be assigned once inside a program.

– Non safety parameters shall not be used to bypass safety functions.

8.3.5 Module test – SW level 1

Each module which was not previously assessed shall be tested against the test cases defined in the module design by functional and black-box, grey-box or white-box testing as appropriate.

If the module does not pass the testing, then predefined corrective action shall be taken.

The test results, and corrective actions, shall be documented.

NOTE 1 Functional testing aims to reveal failures during the specification and design phases, and to avoid failures during implementation and the integration of software and hardware.

NOTE 2 Black-box testing aims to check the dynamic behaviour under real functional conditions, and to reveal failures to meet functional specification, and to assess utility and robustness. Grey-box testing is similar to Black-box testing but additionally monitors relevant test parameter(s) inside the software module.

8.3.6 Software testing – SW level 1

8.3.6.1 General – SW level 1

The main goal of software testing is to ensure that the functionality as detailed in the software design specification is achieved.

The main output of software testing is a document e.g. a test report with test cases and test results allowing an assessment of the test coverage.

Software testing shall also include failure simulation and the associated failure reaction depending on the required safety integrity.

When pre-designed input cards or software modules which incorporate failure detection and reaction are utilised (e.g. discrepancy of input signals or feedback contact of output) then the test of those failure detection and reaction is not necessary. In that case, only the integration of these input cards or software modules in accordance with the manufacturer's specification shall be tested.

Software testing can be carried out as part of the system validation if testing is performed on the target hardware.

Functional testing as a basic measure shall be applied. Code should be tested by simulation where feasible.

It is recommended to define general guidelines or procedures for the testing of safety-related software. These guidelines or procedures should include:

– types of tests to be performed;
– specification of test equipment including tools, support software and configuration description;
– management of software versioning during testing and correcting of safety-related software;
– corrective actions on failed test;
– criteria for the completion of the test with respect to the related functions or requirements; physical location(s) of the testing, such as computer simulation, bench top or lab, factory, or on the machine.

8.3.6.2 Test planning and execution – SW level 1

Test planning based on test cases shall include:

– definition of roles and responsibilities by name;
– installation testing;
– functional testing.

8.3.7 Documentation – SW level 1

All life cycle activities shall be traceable forwards and backwards from the specification of the safety function(s) and through the completed validation plan.

The inputs and outputs of all software safety lifecycle phases shall be documented and made available to the relevant persons.

The test activities results and corrective actions taken shall be documented.

8.3.8 Configuration and modification management process – SW level 1

Any modifications or changes to software shall be subject to an impact analysis that identifies all software parts affected and the necessary re-design, re-review and re-test activities to confirm that the relevant software safety requirements are still satisfied.

Configuration management processes and modifications management processes shall be defined and documented. This shall, as a minimum, include the following items:

– articles managed by the configuration, at least: software safety requirements preliminary and detailed software design, source code modules, plans, procedures and results of the validation tests;
– identification rules which uniquely identify each software module or configuration element;
– modification processes which are comprehensive from request through to implementation.

For each article of configuration, it shall be possible to identify any changes that can have occurred and the versions of any associated elements.

NOTE 1 The purpose is to be able to trace the historical development of each article: what modifications have been made, why, and when.

Software configuration management shall allow a precise and unique software version identification to be obtained. Configuration management should associate all the articles (and their version) needed to demonstrate the functional safety.

All articles in the software configuration shall be covered by the configuration management procedure before being tested or being requested by the analyst for final software version evaluation.

NOTE 2 The objective here is to ensure the evaluation procedure is performed on software with all elements in a precise state. Any subsequent change can necessitate revision of the software so that it can be identifiable by the analyst.

Procedures for the archiving of software and its associated data shall be established (methods for storing backups and archives).

NOTE 3 These backups and archives can be used to maintain and modify software during its functional lifetime.

8.4 Software level 2

8.4.1 Software safety lifecycle – SW level 2

8.4.1.1 Maximum achievable SIL – SW level 2

The maximum achievable SIL for SW level 2 is SIL 2.

8.4.1.2 Software safety lifecycle model – SW level 2

A software safety lifecycle model which is resolved into distinct phases shall be used (e.g. V-model), including management and documentation activities to achieve the required level of safety.

Any software lifecycle model may be used provided all the objectives and requirements of this Subclause 8.4 are met. Safety-related software shall be validated as described in 9.5.4.

Software level 2 is of increased complexity in comparison with SW level 1 due to the use of fully variable programming languages. Therefore, the more detailed V-model in Figure 14 is applicable.

NOTE 1 The V-model is a static model used to structure the software design into small parts. It does not introduce any sequence of creation of specifications or implementation. The left side represents requirements; i.e. things to achieve. The right side details testing of the software.

NOTE 2 On the left side of the V-model, the output of each phase is reviewed. Review means to check the output of a phase in the V-model against the requirements of the input of the same phase. The arrow 'Review' represents the first step of the software verification. Further information on the level of independence of review and testing/verification is available in Annex J.

NOTE 3 Project management techniques and processes can be chosen to be appropriate for the size and scope of the project.

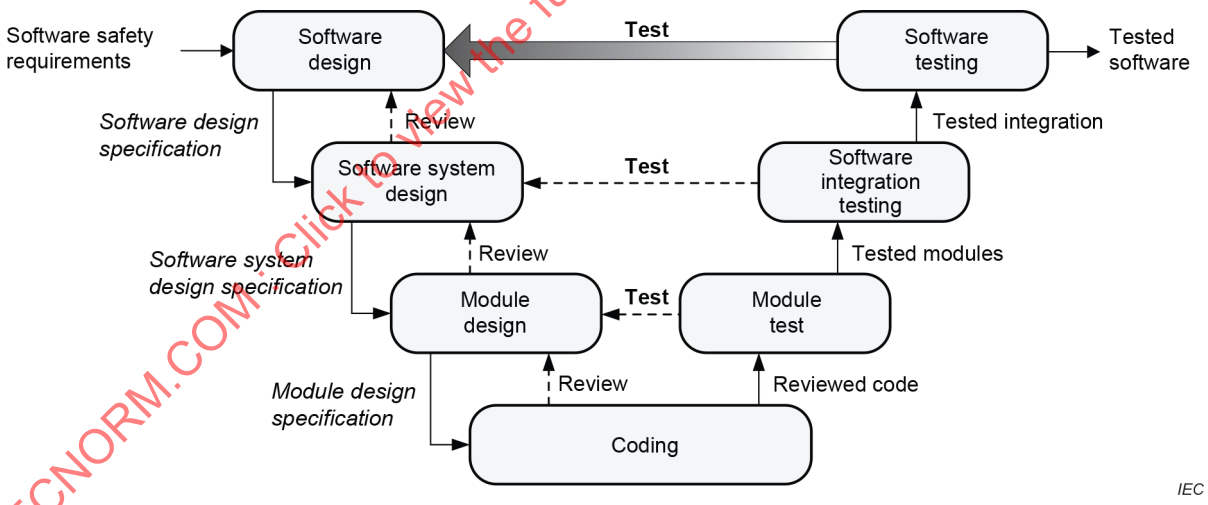


Figure 14 – V-model of software safety lifecycle for SW level 2

NOTE 4 In the V-models, the arrow 'Test' represents the results of test cases according to the specification and, in addition, the need for more precise test case requirements and specifications.

8.4.1.3 Tools usage – SW level 2

A suitable set of tools shall be selected (e.g. configuration management, simulation, and test equipment with test generator). Preferably, the recommended tools from the manufacturer should be applied. The availability of suitable tools for service, updating the machine, and parameterization over the lifetime of the safety-related control system shall be considered. Either the tools provided by the manufacturer of the equipment are used or the suitability of the tools shall be explained and documented.

Suitability shall be proven as follows:

– an analysis carried out to identify possible effects of a failure caused by these tools in the tool chain; and
– appropriate fault avoiding and fault controlling measures be selected, applied, and their effectiveness be verified via rigorous testing and the results documented.

NOTE 1 The appropriateness of fault avoiding and fault controlling measures depends on the severity of the consequence of a failure. The basis of this evaluation is an analysis. To carry out this analysis, it is necessary to have knowledge regarding the application of the support tool and of the machine.

NOTE 2 The effect of failures can vary between different support tools. Therefore IEC 61508-4 differentiates between three categories for off-line support tools used within the software development lifecycle. This can be part of the analysis.

NOTE 3 See IEC 61508-4 for definition of support tools and examples.

NOTE 4 This document does not specify any measures for avoiding or controlling faults of off-line support tools. For examples, see IEC 61508-3:2010, 7.4.4.

8.4.2 Software design – SW level 2

8.4.2.1 General – SW level 2

The software design specification shall be developed on basis of the software safety requirements and managed throughout the lifecycle of the SCS.

8.4.2.2 Software safety requirements – SW level 2

To support the software design process, the following information shall be considered:

a) specification of the safety function(s) (see 5.2);
b) configuration or architecture of the SCS (e.g. hardware architecture, wiring diagram, safety-related inputs and outputs);
c) response time requirements;
d) operator interfaces and controls, such as: switches, joysticks, mode selector, dials, touch sensitive control devices, keypads, etc.;
e) relevant modes of operation of the machine;
f) requirements on diagnostics for hardware including the characteristics of sensors, final actuators, etc.;
g) effects of mechanical tolerances, e.g. of sensors and/or their sensing counter parts;
h) coding guidelines.

When applying SW level 2, the tables of IEC 61508-3:2010, Annex A and Annex B shall be taken into consideration when it is appropriate to use alternative techniques and measures of an equivalent effectiveness. IEC 61508-7 provides additional information.

The design and choice of the language chosen to satisfy the required SIL of the SCS shall be appropriate for the application.

The design shall include self-monitoring of control flow and data flow appropriate to the SIL of the SCS. On failure detection, appropriate actions shall be performed to achieve or maintain a safe state.

8.4.2.3 Software design specification – SW level 2

The software design specification shall be derived from the software safety requirements of the SCS.

The software design specification shall be:

– structured, reviewable, testable, understandable, maintainable and operable;
– developed for each subsystem on the basis of the SCS specification and architecture;
– sufficiently detailed to allow the design and implementation of the SCS to achieve the required level of safety (SIL), and to allow verification and testing.
– traceable back to the specification of the software safety requirements of the SCS. This means that the specification is as such understandable such that another person (e.g. non-software specialist) can verify if the specification corresponds to the software safety requirements of the safety functions defined in the risk assessment.
– free of ambiguous terminology and irrelevant descriptions.

It shall be possible to relate the inputs of the software design specification in a straightforward manner to the desired outputs and vice versa. Where appropriate, easily readable semi-formal methods such as cause&effect tables, logic tables or diagrams, function-blocks or sequence diagrams shall be used in the documentation.

NOTE 1 Where appropriate depends on the number of safety functions involved in the program. Whenever the total amount of safety functions inside the program is larger than 3, it is considered appropriate.

The following shall be specified within the software design specification:

a) logic of the safety functions, including safety-related inputs and outputs and proper diagnostics on detected faults. Possible methods include, but are not limited to, cause&effect table, written description or function blocks;

NOTE 2 Faults can also be detected by hardware (e.g. signal discrepancy detected by input card).

b) test cases that include:
   – the specific input value(s) for which the test is carried out and the expected test results including pass/fail criteria;
   – fault insertion or injection(s).

NOTE 3 For simple functions, the test case(s) can be given implicitly by the specification of the safety function.

c) diagnostic functions for input devices, such as sensing elements and switches, and final control elements, such as solenoids, relays, or contactors;
d) functions that enable the machine to achieve or maintain a safe state;
e) functions related to the detection, annunciation and handling of faults;
f) functions related to the periodic testing of SCS(s) on-line and off-line;
g) functions that prevent unauthorized modification of the SCS (e.g. password);
h) interfaces to non SCS;
i) safety function response time.

NOTE 4 Guidance on software documentation is given in IEC 61508, ISO/IEC/IEEE 26512.

It is recommended to use pre-designed software modules within the software design specification wherever possible.

It is recommended that in case of pre-designed safety sub-functions, for example IEC 61800-5-2, a reference to the specification provided by the manufacturer should be used.

The information in the software design specification shall be reviewed and where necessary revised, to ensure that the requirements of the software safety requirements (see 8.4.2.2) are adequately specified.

8.4.3 Software system design – SW level 2

8.4.3.1 General – SW level 2

Software system design starts with architecture definition. Software architecture shall be established that fulfils the software design specification. The software architecture defines the major elements and subsystems of the software, how they are interconnected and how the required attributes will be achieved. It also defines the overall behavior of the software, and how software elements interface and interact. Examples of major software elements include operating systems, databases, input/output subsystems, communication subsystems, application programs, programming and diagnostic tools, etc.

The software system design shall follow a modular approach with a limited software module size, a fully defined interface and one entry/one exit point in subroutines and functions. Each module shall have a single, clearly understood function or purpose. The maximum module size shall be limited to one complete safety function.

The following programming techniques shall be used to avoid systematic failures:

– range checking and plausibility checking of variables and configuration parameters;
– temporal or logical program sequence monitoring to detect a defective program sequence: A defective program sequence exists if the individual elements of a program (e.g. software modules, subprograms or commands) are processed in the wrong sequence or period of time or if the clock of the processor is faulty (see IEC 61508-7:2010, Clause A.9);
– limiting the number or extent of global variables.

NOTE For Software level 2, see Annex G of IEC 61508-7:2010 for guidance on object oriented architecture and design.

8.4.3.2 Software system design specification – SW level 2

A software system design specification shall be provided as an output of the software system design. This shall explain the main software aspects such as indicated in the following list, for example:

– the software architecture that defines the structure decided to satisfy the software design specification;
– the global data;
– data libraries used;
– pre-existing software modules used;
– diagnostic functions (internal, external);
– programming tools including information which uniquely identifies the tool;
– integration test cases and procedures, including specification of the test environment, support software, configuration description and procedures for corrective action on failure of test.

The information contained in the software system specification shall be reviewed against the software design specification.

8.4.4 Module design – SW level 2

8.4.4.1 General – SW level 2

Where previously developed software library modules are to be used as part of the design, their suitability in satisfying the specification of requirements of the software safety shall be demonstrated. Constraints from the previous software development environment (for example operating system and compiler dependencies) shall be evaluated.

8.4.4.2 Input information – SW level 2

For software modules, the following information shall be available in the software system design specification:

a) module description;
b) module interface (inputs and outputs with data types and, if necessary, with data ranges);
c) module libraries used;
d) special coding rules.

8.4.4.3 Module design specification – SW level 2

The module design specification shall contain the following information:

a) description of the logic (i.e. the functionality) of each module;
b) fully defined input and output interfaces assigned for each module;
c) format and value ranges of input and output data and their relation to modules;
d) test cases which shall include normal and outside normal operation;

NOTE Although test cases usually comprise the individual testing of parameters within their specified ranges, a varying combination of these parameters can introduce unpredicted operation.

e) documentation of the interrupts.

This information shall be reviewed against the input information (see 8.4.4.2).

8.4.5 Coding – SW level 2

Software shall be developed in accordance with the design specifications and coding rules. Coding rules can be either well-known industry standards or can be internal to the manufacturer. The code shall be reviewed against the design specifications and coding rules.

NOTE 1 Coding rules are intended to restrict the freedom of programming in order to avoid the program code becoming incomprehensible and in order to reduce the likelihood of the program entering unintended states.

NOTE 2 Coding rules usually define a subset of a programming language or use of a strongly typed programming language (see IEC 61508-7:2010, C.4.1).

The output of coding shall comprise

– source code listing (e.g. ladder, function blocks, models);
– code review report.

Some typical coding rules to be applied, include, but are not limited to the following:

– Structure of the program is as easy and clear as possible.
– Structure of the program should be such that the logical flow starts at the top and follows in the effective sequence.
– Every part should have sufficient comments in a predefined way.
– Same names for parameters as during design should be used.
– Names should represent the function of the parameter in a clear way.
– A predefined state should exist.
– Use of set/reset for safety functions should be limited.
– Safety outputs should only be assigned once inside a program.
– Non safety parameters shall not be used to bypass safety functions.

8.4.6 Module test – SW level 2

Each module which was not previously assessed shall be tested against the test cases defined in the module design by functional and black-box, grey-box or white-box testing as appropriate.

If the module does not pass the testing, then predefined corrective action shall be taken.

The test results and corrective actions shall be documented.

NOTE 1 Functional testing aims to reveal failures during the specification and design phases, and to avoid failures during implementation and the integration of software and hardware.

NOTE 2 Black-box testing aims to check the dynamic behaviour under real functional conditions, and to reveal failures to meet functional specification, and to assess utility and robustness. Grey-box testing is similar to Black-box testing but additionally monitors relevant test parameter(s) inside the software module.

Module testing shall use as a minimum dynamic analysis and testing.

8.4.7 Software integration testing SW level 2

The software shall be tested against the integration test cases. The results of software integration testing shall be documented.

NOTE The objective of these tests is to show that all software modules and software elements/subsystems interact correctly to perform their intended function and do not perform unintended functions. This does not imply testing of all input combinations, nor of all output combinations. Testing all equivalence classes or structure based testing can be sufficient. Boundary value analysis or control flow analysis can reduce the test cases to an acceptable number. Analysable programs make the requirements easier to fulfil.

8.4.8 Software testing SW level 2

8.4.8.1 General – SW level 2

The main goal of software testing is to ensure that the functionality as detailed in the software design specification is achieved.

NOTE This can imply testing of all input combinations, and/or all output combinations.

The main output of software testing is a document, e.g. a test report with test cases and test results allowing an assessment of the test coverage.

Software testing shall also include failure simulation and the associated failure reaction depending on the required safety integrity.

When pre-designed input cards or software modules which incorporate failure detection and reaction are utilised (e.g. discrepancy of input signals or feedback contact of output) then the test of those failure detection and reaction is not necessary. In that case, only the integration of these input cards or software modules in accordance with the manufacturer's specification shall be tested.

Software testing can be carried out as part of the system validation if testing is performed on the target hardware.

Functional testing as a basic measure shall be applied. Code should be tested by simulation where feasible.

It is recommended to define general guidelines or procedures for the testing of safety-related software. These guidelines or procedures should include:

– types of tests to be performed;
– specification of test equipment including tools, support software and configuration description;
– management of software versioning during testing and correcting of safety-related software;
– corrective actions on failed test;
– criteria for the completion of the test with respect to the related functions or requirements; physical location(s) of the testing, such as computer simulation, bench top or lab, factory, or on the machine.

8.4.8.2 Test planning and execution – SW level 2

Test planning based on test cases shall include:

– definition of roles and responsibilities by name;
– installation testing;
– functional testing.

Testing of software includes two types of activities:

– Static analysis: Analysis of software documentation, e.g. by review, inspection, walk-through, control flow analysis, or dataflow analysis.
– Dynamic testing: Execution of the software in a controlled and systematic way, so as to demonstrate the presence of the required behaviour and the absence of unwanted behaviour. This includes, in particular, functional testing, black-box or grey-box-testing.

In the early phases of the software lifecycle, verification is static. Dynamic testing becomes possible when code is produced. For verifying the output of software lifecycle activities, both activities are required in combination. For further description of static analysis and dynamic testing, see IEC 61508-3.

The following is required for verification and testing of safety-related software:

– static analysis shall be done and documented in any case;
– dynamic testing shall be done and documented;
– where software is required for a safety function of up to SIL 1 and is not subject to dynamic testing, this shall be justified with respect to the structural simplicity of the software;
– for dynamic testing, every subprogram (subroutine or function) shall have been called at least once (entry points) during testing;
– for software which is required for a safety function of SIL 2, all statements in the code shall be executed at least once during dynamic testing;
– where software is used in diagnostic functions for controlling random hardware failures, dynamic testing shall address the correct implementation of the diagnostics, e.g. by fault insertion testing;
– dynamic testing shall include a final test on the target hardware.

8.4.9 Documentation – SW level 2

All life cycle activities shall be traceable forwards and backwards from the specification of the safety function(s) and through the completed validation plan.

The inputs and outputs of all software safety lifecycle phases shall be documented and made available to the relevant persons.

The test activities results and corrective actions taken shall be documented.

8.4.10 Configuration and modification management process – SW level 2

Any modifications or changes to software shall be subject to an impact analysis that identifies all software parts affected and the necessary re-design, re-review and re-test activities to confirm that the relevant software safety requirements are still satisfied.

Configuration management processes and modifications management processes shall be defined and documented. This shall, as a minimum, include the following items:

– articles managed by the configuration, at least: software safety requirements, preliminary and detailed software design, source code modules, plans, procedures and results of the validation tests;
– identification rules which uniquely identify each software module or configuration element;
– modification processes which are comprehensive from request through to implementation.

For each article of configuration, it shall be possible to identify any changes that can have occurred and the versions of any associated elements.

NOTE 1 The purpose is to be able to trace the historical development of each article: what modifications have been made, why, and when.

Software configuration management shall allow a precise and unique software version identification to be obtained. Configuration management shall associate all the articles (and their version) needed to demonstrate the functional safety.

All articles in the software configuration shall be covered by the configuration management procedure before being tested or being requested by the analyst for final software version evaluation.

NOTE 2 The objective here is to ensure that the evaluation procedure be performed on software with all elements in a precise state. Any subsequent change can necessitate revision of the software so that it can be identifiable by the analyst.

Procedures for the archiving of software and its associated data shall be established (methods for storing backups and archives).

NOTE 3 These backups and archives can be used to maintain and modify software during its functional lifetime.

9 Validation

9.1 Validation principles

In this document, the purpose of the validation is to confirm that the SCS complies with the safety requirements specification given in Clause 5 and the information for use in 10.3.

NOTE 1 In this document, the validation is limited to the designed SCS or a part of it supporting the safety functions required from the risk reduction strategy at the machine level given in ISO 12100. The SCS validation result is intended to be part of the overall validation of the machine.

NOTE 2 In some cases, the safety validation can only be completed after final installation (for example, when the application software development is not finalized).

The validation activities consist of collecting and checking the availability of the evidence demonstrating the completeness of each design activity identified in the safety plan.

The validation to be applied to the SCS includes inspection (e.g. by analysis) and testing of the SCS to ensure that it achieves the requirements stated in the safety requirements specification (according to Clause 5).

The validation shall demonstrate that the SCS meets the requirements and, in particular, the following:

a) the specified functional requirements of the safety functions provided by that part (see 5.2), as set out in the design rationale;

b) the requirements of the specified SIL.

Validation shall be carried out by persons who are independent from the design of the SCS.

NOTE 3 "Independent person" does not necessarily mean that a third-party test is required.

The analysis should be started as early as possible in, and in parallel with, the design process.

NOTE 4 Problems can then be corrected early while they are still relatively easy to correct, i.e. during steps "design and technical realization of the safety function" and "evaluate the SIL". It can be necessary for some parts of the analysis to be delayed until the design is well developed.

Figure 15 gives an overview of the validation process: validation consists of applying analysis (see 9.2) and executing functional tests (see 9.3) under foreseeable conditions in accordance with the validation plan. The balance between the analysis and testing shall be justified. For architectures with diagnostic function, the validation of the safety function shall also include testing under fault conditions to show that the fault reaction will be initiated by the implemented diagnostic function.

Where appropriate due to the system's size, complexity or the effects of integrating it with the control system (of the machinery), special arrangements should be made for

– validation of the subsystem separately before integration, including simulation of the appropriate input and output signals, and

– validation of the effects of integrating safety-related parts into the remainder of the control system within the context of its use in the machine.

"Modification of the design" in Figure 15 refers to the design process. If the validation cannot be successfully completed, changes in the design are necessary. The validation of the SCS should then be repeated as appropriate. This process should be iterated until the SCS for each safety function is successfully validated.

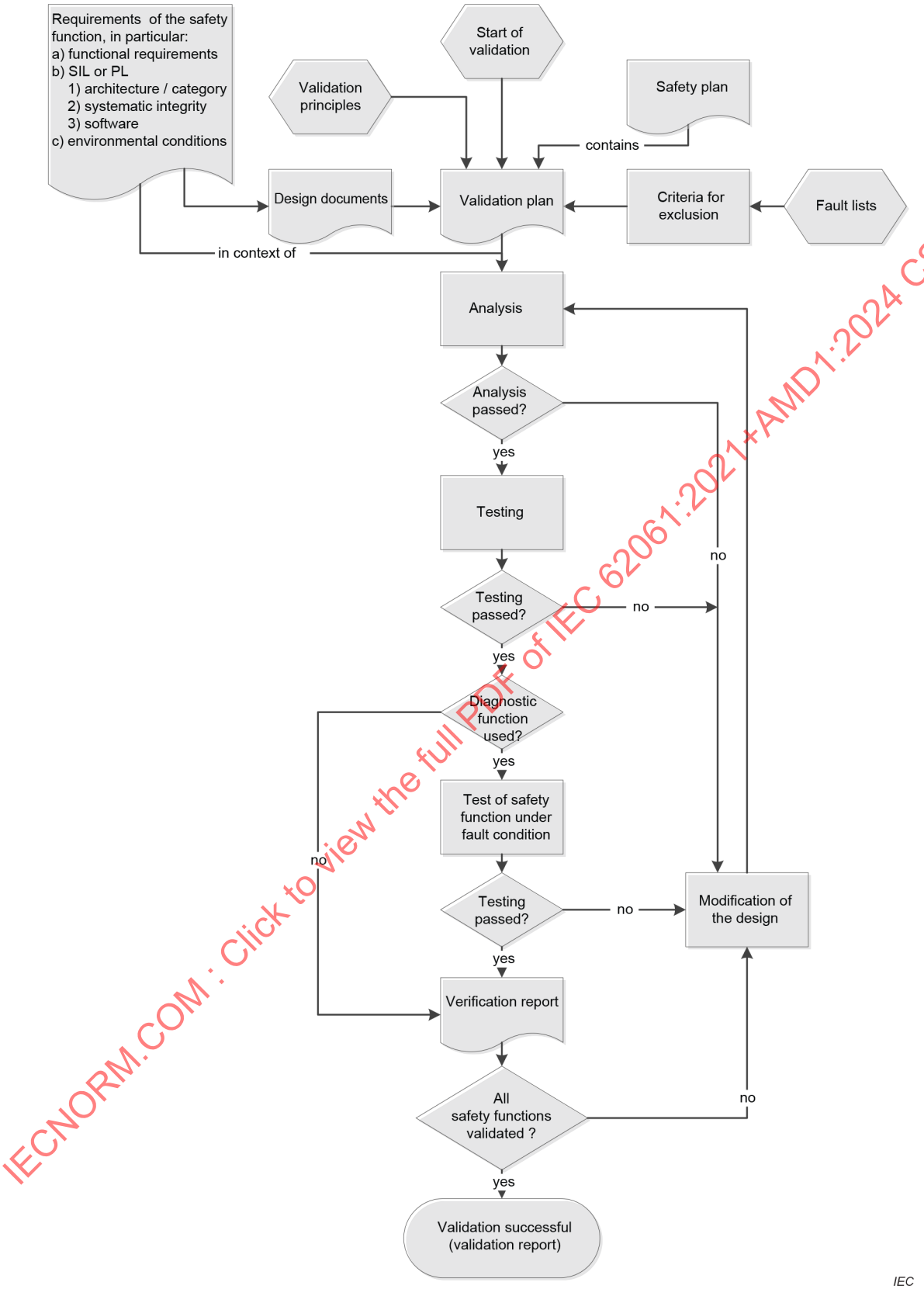


Figure 15 – Overview of the validation process

9.1.1 Validation plan

The validation plan shall identify and describe the requirements for carrying out the validation process. The validation plan shall also identify the means to be employed to validate the specified safety functions. It shall set out, where appropriate:

a) the identity of the specification documents,
b) the operational and environmental conditions during testing,
c) the analyses and tests to be applied,
d) the reference to test standards to be applied,
e) the persons or parties responsible for each step in the validation process, and
f) the required equipment.

Subsystems which have previously been validated to the same specification need only reference to that previous validation

NOTE Information on the level of independence of validation is available in Annex J.

9.1.2 Use of generic fault lists

Validation involves consideration of the behaviour of the SCS for all faults to be considered. A basis for fault consideration is given in the tables of fault lists of ISO 13849-2:2012, Annexes A to D, which are based on experience and which contain:

– the components/elements to be included, e.g. conductors/cables,
– the faults to be taken into account, e.g. short circuits between conductors,
– the permitted fault exclusions, taking into account environmental, operating and application aspects, and
– a remarks section giving the reasons for the fault exclusions.

Only permanent faults are taken into account in the fault lists.

9.1.3 Specific fault lists

If necessary, a specific product-related fault list shall be generated as a reference document for the validation of the subsystem(s) and/or subsystem element(s).

NOTE The list can be based on the appropriate generic list(s) found in the annexes A to D in ISO 13849-2:2012.

Where the specific product-related fault list is based on the generic list(s), it shall state

a) the faults taken from the generic list(s) to be included,
b) any other relevant faults to be included but not given in the generic list (e.g. common-cause failures),
c) the faults taken from the generic list(s) which may be excluded on the basis that the criteria given in the generic list(s) are satisfied (see 7.3.3),
d) and exceptionally any other faults for which the generic list(s) do not permit an exclusion, but for which justification and rationale for an exclusion is presented (see 7.3.3).

Where this list is not based on the generic list(s), the designer shall give the rationale for fault exclusions.

9.1.4 Information for validation

The information required for validation will vary with the technology used, the architectural constraints and SIL to be demonstrated, the design rationale of the system, and the contribution of the SCS to the reduction of the risk. Documents containing sufficient information from the following list shall be included as appropriate in the validation to demonstrate that the safety-related parts perform the specified safety functions to the required SIL and architectural constraints:

a) specification of the required characteristics of each safety function, especially the required SIL and architectural constraints;

b) drawings and specifications, e.g. for mechanical, hydraulic and pneumatic parts, printed circuit boards, assembled boards, internal wiring, enclosure, materials, mounting;

c) block diagram(s) with a functional description of the blocks;

d) circuit diagram(s), including interfaces/connections;

e) functional description of the circuit diagram(s);

f) time sequence diagram(s) for switching components, signals relevant for safety;

g) description of the relevant characteristics of components previously validated;

h) for safety-related parts other than those listed in g), component lists with item designations, rated values, tolerances, relevant operating stresses, type designation, failure-rate data and component manufacturer, and any other data relevant to safety;

i) analysis of all relevant faults according to 9.1.2 and 9.1.3, such as those listed in the tables of ISO 13849-2:2012, Annexes A to D, including the justification of any excluded faults;

j) an analysis of the influence of processed materials;

k) information for use, e.g. installation and operation manual/instruction handbook.

Where software is relevant to the safety function(s), the software documentation shall include

– a specification which is clear and unambiguous and which states the safety integrity the software is required to achieve,

– evidence that the software is designed to achieve the required SIL (see 9.5.4), and

– details of the verification (in particular test reports) carried out to prove that the required SIL is achieved.

Information is required on how the SIL and PFH is determined. The documentation of the quantifiable aspects shall include

– the basic subsystem architecture according to 7.5.2,

– the determination reliability parameters (e.g. $MTTF_D$ or λ_D of subsystem elements and CCF), and

– the determination of the architectural constraints.

Information is required for documentation on systematic aspects of the SCS. Information is required to describe how the combination of several subsystems achieves a SIL in accordance with the required SIL.

9.1.5 Validation record

Validation by analysis and testing shall be recorded, see also Clause 10. Appropriate documentation shall state:

- the version of the validation plan being used, and the version of the safety function tested;
- the safety function under test (or analysis), along with the specific reference to the requirement specified during the validation planning;
- referenced standards;

- tools and equipment used, along with calibration data;
- the results of each test;
- discrepancies between expected and actual results.

9.2 Analysis as part of validation

9.2.1 General

Validation of the SCS shall be carried out by analysis. Inputs to the analysis include the following:

– the safety function(s), their characteristics and the safety integrity specified according to Clause 5;

– the system structure (e.g. basic subsystem architectures) according to 7.5.2;

– the quantifiable aspects (e.g. $MTTF_D$ or λ_D, DC and CCF) according to 6.4.2;

– the non-quantifiable, qualitative aspects which affect system behaviour (if applicable, software aspects);

– deterministic arguments.

NOTE 1 A deterministic argument is an argument based on qualitative aspects (e.g. quality of manufacture, experience of use). This consideration depends on the application, which, together with other factors, can affect the deterministic arguments.

NOTE 2 Deterministic arguments differ from other evidence in that they show that the required properties of the system follow logically from a model of the system. Such arguments can be constructed on the basis of simple, well-understood concepts.

9.2.2 Analysis techniques

The selection of an analysis technique depends upon the particular object. Two basic techniques exist, as follows.

a) Top-down (deductive) techniques are suitable for determining the initiating events that can lead to identified top events, and calculating the probability of top events from the probability of the initiating events. They can also be used to investigate the consequences of identified multiple faults.

EXAMPLE Fault tree analysis (FTA, see IEC 61025), event tree analysis (ETA, see IEC 62502).

b) Bottom-up (inductive) techniques are suitable for investigating the consequence of identified single faults.

EXAMPLE Failure modes and effects analysis (FMEA, see IEC 60812) and failure modes, effects and criticality analysis (FMECA).

9.2.3 Verification of safety requirements specification (SRS)

The requirements specification for the safety function shall be verified to ensure consistency and completeness and correctness for its intended use.

Verification may be performed by reviews and inspections of the SCS safety requirements and design specification(s), in particular to prove that all aspects of

– the intended application requirements and safety needs, and

– the operational and environmental conditions and possible human errors (e.g. misuse) have been considered.

9.3 Testing as part of validation

9.3.1 General

Testing shall be carried out to complete the validation. Validation tests shall be planned and implemented in a logical manner. In particular:

a) a test plan shall be produced before testing begins that shall include
   1) the test specifications;
   2) the required outcome of the tests for compliance, and
   3) the chronology of the tests;
b) test records shall be produced that include
   4) the name of the person carrying out the test;
   5) the environmental conditions;
   6) the test procedures and equipment used;
   7) the date of the test, and
   8) the results of the test;
c) the test records shall be compared with the test plan to ensure that the specified functional and performance targets are achieved.

The test sample shall be operated as near as possible to its final operating configuration.

This testing can be applied manually or automatically, e.g. by computer.

Where applied, validation of the safety functions by testing shall be carried out by applying input signals, in various combinations, to the SCS. The resultant response at the outputs shall be compared to the appropriate specified outputs.

It is recommended that the combination of these input signals be applied systematically to the control system and the machine. An example of this logic is power-on, start-up, operation, directional changes, restart-up. Where necessary, an expanded range of input data shall be applied to take into account anomalous or unusual situations, in order to see how the SCS responds. Such combinations of input data shall take into account foreseeable incorrect operation(s).

The objectives of the test will determine the environmental condition for that test, which can be one or another of the following:

– the environmental conditions of intended use;
– the conditions at a particular rating;
– a given range of conditions if drift is expected.

9.3.2 Measurement accuracy

The accuracy of measurements during the validation by testing shall be appropriate for the test carried out. In general, these measurement accuracies shall be within 5 K for temperature measurements and 5 % for the following:

a) time measurements;
b) pressure measurements;
c) force measurements;
d) electrical measurements;
e) relative humidity measurements;

f) linear measurements.

Deviations from these measurement accuracies shall be justified.

9.3.3 More stringent requirements

If, according to its accompanying documentation, the requirements for the SCS exceed those within this document, the more stringent requirements shall apply.

NOTE More stringent requirements can apply if the control system has to withstand particularly adverse service conditions, e.g. rough handling, humidity effects, hydrolysation, ambient temperature variations, effects of chemical agents, corrosion, high strength of electromagnetic fields – for example, due to close proximity of transmitters.

9.3.4 Test samples

Test samples shall not be modified during the course of the tests.

Certain tests can permanently change the performance of some components. Where a permanent change in a component causes the safety-related part to be incapable of meeting the requirements of further tests, a new sample or samples shall be used for subsequent tests.

Where a particular test is destructive and equivalent results can be obtained by testing part of SCS in isolation, a sample of that SCS may be used instead of the whole SCS for the purpose of obtaining the results of the test. This approach shall only be applied where it has been shown by analysis that testing of a part of SCS is sufficient to demonstrate the safety integrity of the whole SCS that performs the safety function.

9.4 Validation of the safety function

9.4.1 General

The validation of safety functions shall demonstrate that the SCS provides the safety function(s) in accordance with their specified characteristics.

NOTE 1 A loss of the safety function in the absence of a hardware fault is due to a systematic fault, which can be caused by errors made during the design and integration stages (a misinterpretation of the safety function characteristics, an error in the logic design, an error in hardware assembly, an error in typing the code of software, etc.). Some of these systematic faults will be revealed during the design process, while others will be revealed during the validation process or will remain unnoticed. In addition, it is also possible for an error to be made (e.g. failure to check a characteristic) during the validation process.

Validation of the specified characteristics of the safety functions shall be achieved by the application of appropriate measures from the following list:

– functional analysis of schematics, reviews of the software (see 9.5.3);

NOTE 2 Where a machine has complex or a large number of safety functions, an analysis can reduce the number of functional tests required.

– simulation;
– check of the hardware components installed in the machine and details of the associated software to confirm their correspondence with the documentation (e.g. manufacture, type, version);
– functional testing of the safety functions in all required operating modes as defined in the SRS of the machine, to establish whether they meet the specified characteristics (see Clause 5). The functional tests shall ensure that all safety-related outputs are realized over their complete ranges and respond to safety-related input signals in accordance with the specification. The test cases are normally derived from the specifications but could also include some cases derived from analysis of the schematics or software;
– extended functional testing to check foreseeable abnormal signals or combinations of signals from any input source, including power interruption and restoration, and incorrect operations;

– check usability of the operator–SCS interface.

NOTE 3 Consider for example an HMI for software based parameterization of the safety function. In general, more information is available in IEC 60204-1 or IEC 61310 (all parts).

NOTE 4 Other measures against systematic failures mentioned in 9.5.2 (e.g. diversity, failure detection by automatic tests) can also contribute in the detection of functional faults.

9.4.2 Analysis and testing

Analysis and testing will require failure analysis using circuit diagrams and, where the failure analysis does not reach a clear result:

– fault injection tests on the actual circuit and fault initiation on actual components, particularly in parts of the system where there is doubt regarding the results obtained from failure analysis (see 9.2); or

– a simulation of control system behaviour in the event of a fault, e.g. by means of hardware and/or software models.

Fault injection or fault simulation tests can be performed at different levels, e.g. subsystem element or subsystem level, considering the specific application and test setup.

When validating by testing, the tests shall include, as appropriate,

– fault injection tests into a production sample,

– fault injection tests into a hardware model,

– software simulation of faults, and

– subsystem failure, e.g. power supplies.

The precise instant at which a fault is injected into a system can be critical. The worst-case effect of a fault injection shall be determined by analysis and by injecting the fault at this appropriate critical time.

9.5 Validation of the safety integrity of the SCS

9.5.1 General

The following steps shall be performed:

– verification for correct evaluation of SIL of the SCS based on subsystems, architecture and reliability parameters (e.g. DC and $MTTF_D$ or λ_D);

– verification that the SIL achieved by the SCS satisfies the required SIL in the safety requirements specification for the machinery: SIL ≥ required SIL.

9.5.2 Validation of subsystem(s)

The safety integrity of each subsystem of the SCS is characterized by its SIL and shall be validated by confirming (verification) the following:

– the used architecture (see 7.5.2), and

– the PFH (see 7.6), and

– the systematic integrity (see 7.3.2, Software, CCF).

In this context, the validation of $MTTF_D$ or λ_D, DC and CCF is typically performed by analysis and visual inspection. The $MTTF_D$ or λ_D values for components (including B_{10} or B_{10D}, T_{10D} and duty cycle values) shall be checked for plausibility. For example, the value given on the manufacturer's datasheet is to be compared with Annex A.

NOTE 1 A fault exclusion implies infinite $MTTF_D$; therefore, the component will not contribute to the calculation of channel $MTTF_D$.

The DC values for components (subsystem elements) and/or logic blocks shall be checked for plausibility (e.g. against measures in Annex D). The correct implementation (hardware and software) of checks and diagnostics, including appropriate fault reaction, shall be validated by testing under typical environmental conditions in use.

The correct implementation of sufficient measures against common-cause failures shall be validated (e.g. against Annex E). Typical validation measures are static hardware analysis and functional testing under environmental conditions.

NOTE 2 Generally, for the specification of the $MTTF_D$ or λ_D values of electronic components, an ambient temperature of +40 °C is taken as a basis. During validation, it is important to ensure that, for $MTTF_D$ or λ_D values, the environmental and functional conditions (in particular temperature) taken as basis are met. Where a device, or component, is operated significantly above (e.g. more than 15 °C) the specified temperature of +40 °C, it will be necessary to use $MTTF_D$ or λ_D values for the increased ambient temperature.

9.5.3 Validation of measures against systematic failures

The validation of measures against systematic failures can typically be provided by:

a) inspections of design documents which confirm the application of
   - basic and well-tried safety principles (see ISO 13849-2:2012, Annexes A to D);
   - further measures for avoidance of systematic failures, and
   - further measures for the control of systematic failures such as hardware diversity, modification protection or failure assertion programming;

b) failure analysis (e.g. FMEA);

c) fault injection tests/fault initiation;

d) inspection and testing of data communication, e.g. parameterization, installation;

e) checking that a quality management system avoids the causes of systematic failures in the manufacturing process.

9.5.4 Validation of safety-related software

The validation of software shall include:

– the specified functional behaviour and performance criteria (e.g. timing performance) of the software when executed on the target hardware,

– verification that the software measures are sufficient for the specified required SIL of the safety function, and

– measures and activities taken during software development to avoid systematic software faults.

As a first step, check that there is documentation for the specification and design of the safety-related software. This documentation shall be reviewed for completeness and absence of erroneous interpretations, omissions or inconsistencies.

NOTE In the case of small programs, an analysis of the program by means of reviews or walk-through of control flow, procedures, etc. using the software documentation (control flow chart, source code of modules or blocks, I/O and variable allocation lists, cross-reference lists) can be sufficient.

In general, software can be considered a “black box” or “grey box” (see Clause 8), and validated by the black- or grey-box test, respectively.

Depending on the required SIL, the tests should include, as appropriate,

– black-box testing of functional behaviour and performance (e.g. timing performance),

– additional extended test cases based upon limit value analyses, recommended for SIL 2 or SIL 3,
– I/O tests to ensure that the safety-related input and output signals are used properly, and
– test cases which simulate faults determined analytically beforehand, together with the expected response, in order to evaluate the adequacy of the software-based measures for control of failures.

Individual software functions which have already been validated do not need to be validated again. Where a number of such safety function blocks are combined for a specific project, however, the resulting total safety function shall be validated.

Software documentation shall be checked to confirm that sufficient measures and activities have been implemented against systematic software faults in accordance with the simplified V-model (see Figure 12).

The measures for software implementation and configuration and modification management according to Clause 8, which depend on the SIL to be attained, shall be examined with regard to their proper implementation.

Should the safety-related software be subsequently modified, it shall be revalidated on an appropriate scale.

9.5.5 Validation of combination of subsystems

Where the safety function is implemented by two or more subsystems, validation of the combination – by analysis and, if necessary, by testing – shall be undertaken to establish that the combination achieves the safety integrity specified in the design. Existing recorded validation results of subsystems can be taken into account. The following validation steps shall be performed:

– inspection of design documents describing the overall safety function(s);
– a check that the overall SIL of the subsystem combination has been correctly evaluated, based on the SIL of each individual subsystem (according to 6.4.2);
– consideration of the characteristics of the interfaces, e.g. voltage, current, pressure, data format of information, signal level;
– failure analysis relating to combination/integration, e.g. by FMEA;
– for redundant systems, fault injection tests relating to combination/integration.

10 Documentation

10.1 General

The manufacturer of an SCS and the manufacturer of subsystems shall prepare the relevant technical documentation in 10.2 and information for use in 10.3.

The documentation shall demonstrate the procedure that has been followed and the results that have been received.

The documentation shall be subject to version control.

10.2 Technical documentation

The documentation shall contain information relevant to the safety-related part:

– safety function(s) provided by the SCS according to Clause 5 or the safety sub-function provided by the SCS subsystem;