

INTERNATIONAL STANDARD

NORME INTERNATIONALE



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

Sécurité des machines – Sécurité fonctionnelle des systèmes de commande relatifs à la sécurité

IECNORM.COM : Click to view the full PDF of IEC 62061:2021



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2021 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office
3, rue de Varembe
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 It also gives information on projects, replaced and withdrawn publications.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 online collection - oc.iec.ch

Discover our powerful search engine and read freely all the publications previews. 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 22 000 terminological entries in English and French, with equivalent terms in 18 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Recherche de publications IEC -

webstore.iec.ch/advsearchform

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études, ...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et une fois par mois par email.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: sales@iec.ch.

IEC online collection - oc.iec.ch

Découvrez notre puissant moteur de recherche et consultez gratuitement tous les aperçus des publications. Avec un abonnement, vous aurez toujours accès à un contenu à jour adapté à vos besoins.

Electropedia - www.electropedia.org

Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 000 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



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

Sécurité des machines – Sécurité fonctionnelle des systèmes de commande relatifs à la sécurité

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 13.110; 25.040.99; 29.020

ISBN 978-2-8322-9333-1

Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.

CONTENTS

FOREWORD.....	8
INTRODUCTION.....	10
1 Scope.....	11
2 Normative references	12
3 Terms, definitions and abbreviations	13
3.1 Alphabetical list of definitions.....	13
3.2 Terms and definitions.....	15
3.3 Abbreviations.....	28
4 Design process of an SCS and management of functional safety.....	28
4.1 Objective	28
4.2 Design process	29
4.3 Management of functional safety using a functional safety plan.....	31
4.4 Configuration management	33
4.5 Modification	33
5 Specification of a safety function	34
5.1 Objective	34
5.2 Safety requirements specification (SRS).....	34
5.2.1 General	34
5.2.2 Information to be available.....	34
5.2.3 Functional requirements specification.....	35
5.2.4 Estimation of demand mode of operation	35
5.2.5 Safety integrity requirements specification.....	36
6 Design of an SCS	37
6.1 General.....	37
6.2 Subsystem architecture based on top down decomposition	37
6.3 Basic methodology – Use of subsystem	37
6.3.1 General	37
6.3.2 SCS decomposition	38
6.3.3 Sub-function allocation	39
6.3.4 Use of a pre-designed subsystem.....	39
6.4 Determination of safety integrity of the SCS.....	40
6.4.1 General	40
6.4.2 PFH.....	40
6.5 Requirements for systematic safety integrity of the SCS	41
6.5.1 Requirements for the avoidance of systematic hardware failures	41
6.5.2 Requirements for the control of systematic faults.....	42
6.6 Electromagnetic immunity	43
6.7 Software based manual parameterization.....	43
6.7.1 General	43
6.7.2 Influences on safety-related parameters	43
6.7.3 Requirements for software based manual parameterization	44
6.7.4 Verification of the parameterization tool.....	45
6.7.5 Performance of software based manual parameterization	45
6.8 Security aspects	45
6.9 Aspects of periodic testing	46
7 Design and development of a subsystem.....	46

7.1	General.....	46
7.2	Subsystem architecture design	47
7.3	Requirements for the selection and design of subsystem and subsystem elements	48
7.3.1	General	48
7.3.2	Systematic integrity	48
7.3.3	Fault consideration and fault exclusion	51
7.3.4	Failure rate of subsystem element	52
7.4	Architectural constraints of a subsystem	55
7.4.1	General	55
7.4.2	Estimation of safe failure fraction (<i>SFF</i>)	56
7.4.3	Behaviour (of the SCS) on detection of a fault in a subsystem	57
7.4.4	Realization of diagnostic functions	58
7.5	Subsystem design architectures	59
7.5.1	General	59
7.5.2	Basic subsystem architectures	59
7.5.3	Basic requirements	61
7.6	<i>PFH</i> of subsystems	62
7.6.1	General	62
7.6.2	Methods to estimate the <i>PFH</i> of a subsystem	62
7.6.3	Simplified approach to estimation of contribution of common cause failure (CCF)	62
8	Software	62
8.1	General	62
8.2	Definition of software levels	63
8.3	Software – Level 1	64
8.3.1	Software safety lifecycle – SW level 1	64
8.3.2	Software design – SW level 1	65
8.3.3	Module design – SW level 1	67
8.3.4	Coding – SW level 1	67
8.3.5	Module test – SW level 1	68
8.3.6	Software testing – SW level 1	68
8.3.7	Documentation – SW level 1	69
8.3.8	Configuration and modification management process – SW level 1	69
8.4	Software level 2	70
8.4.1	Software safety lifecycle – SW level 2	70
8.4.2	Software design – SW level 2	71
8.4.3	Software system design – SW level 2	73
8.4.4	Module design – SW level 2	73
8.4.5	Coding – SW level 2	74
8.4.6	Module test – SW level 2	75
8.4.7	Software integration testing SW level 2	75
8.4.8	Software testing SW level 2	75
8.4.9	Documentation – SW level 2	76
8.4.10	Configuration and modification management process – SW level 2	77
9	Validation	77
9.1	Validation principles	77
9.1.1	Validation plan	80
9.1.2	Use of generic fault lists	80

9.1.3	Specific fault lists	80
9.1.4	Information for validation	81
9.1.5	Validation record	81
9.2	Analysis as part of validation	82
9.2.1	General	82
9.2.2	Analysis techniques	82
9.2.3	Verification of safety requirements specification (SRS)	82
9.3	Testing as part of validation	83
9.3.1	General	83
9.3.2	Measurement accuracy	83
9.3.3	More stringent requirements	84
9.3.4	Test samples	84
9.4	Validation of the safety function	84
9.4.1	General	84
9.4.2	Analysis and testing	85
9.5	Validation of the safety integrity of the SCS	85
9.5.1	General	85
9.5.2	Validation of subsystem(s)	85
9.5.3	Validation of measures against systematic failures	86
9.5.4	Validation of safety-related software	86
9.5.5	Validation of combination of subsystems	87
10	Documentation	87
10.1	General	87
10.2	Technical documentation	87
10.3	Information for use of the SCS	89
10.3.1	General	89
10.3.2	Information for use given by the manufacturer of subsystems	89
10.3.3	Information for use given by the SCS integrator	90
Annex A (informative)	Determination of required safety integrity	92
A.1	General	92
A.2	Matrix assignment for the required SIL	92
A.2.1	Hazard identification/indication	92
A.2.2	Risk estimation	92
A.2.3	Severity (Se)	93
A.2.4	Probability of occurrence of harm	93
A.2.5	Class of probability of harm (CI)	96
A.2.6	SIL assignment	96
A.3	Overlapping hazards	98
Annex B (informative)	Example of SCS design methodology	99
B.1	General	99
B.2	Safety requirements specification	99
B.3	Decomposition of the safety function	99
B.4	Design of the SCS by using subsystems	100
B.4.1	General	100
B.4.2	Subsystem 1 design – “guard door monitoring”	100
B.4.3	Subsystem 2 design – “evaluation logic”	102
B.4.4	Subsystem 3 design – “motor control”	103
B.4.5	Evaluation of the SCS	103
B.4.6	PFH	104

B.5	Verification.....	104
B.5.1	General	104
B.5.2	Analysis.....	104
B.5.3	Tests	105
Annex C (informative)	Examples of $MTTF_D$ values for single components	106
C.1	General.....	106
C.2	Good engineering practices method	106
C.3	Hydraulic components.....	106
C.4	$MTTF_D$ of pneumatic, mechanical and electromechanical components	107
Annex D (informative)	Examples for diagnostic coverage (DC).....	109
Annex E (informative)	Methodology for the estimation of susceptibility to common cause failures (CCF).....	111
E.1	General.....	111
E.2	Methodology	111
E.2.1	Requirements for CCF	111
E.2.2	Estimation of effect of CCF	111
Annex F (informative)	Guideline for software level 1	114
F.1	Software safety requirements.....	114
F.2	Coding guidelines	115
F.3	Specification of safety functions	116
F.4	Specification of hardware design	117
F.5	Software system design specification.....	119
F.6	Protocols	121
Annex G (informative)	Examples of safety functions.....	124
Annex H (informative)	Simplified approaches to evaluate the PFH value of a subsystem	125
H.1	Table allocation approach	125
H.2	Simplified formulas for the estimation of PFH	127
H.2.1	General	127
H.2.2	Basic subsystem architecture A: single channel without a diagnostic function	127
H.2.3	Basic subsystem architecture B: dual channel without a diagnostic function	128
H.2.4	Basic subsystem architecture C: single channel with a diagnostic function	128
H.2.5	Basic subsystem architecture D: dual channel with a diagnostic function(s)	133
H.3	Parts count method.....	134
Annex I (informative)	The functional safety plan and design activities	135
I.1	General.....	135
I.2	Example of a machine design plan including a safety plan	135
I.3	Example of activities, documents and roles	135
Annex J (informative)	Independence for reviews and testing/verification/validation activities	138
J.1	Software design	138
J.2	Validation.....	138
Bibliography		140

Figure 2 – Integration within the risk reduction process of ISO 12100 (extract)	29
Figure 3 – Iterative process for design of the safety-related control system	30
Figure 4 – Example of a combination of subsystems as one SCS.....	31
Figure 5 – By activating a low demand safety function at least once per year it can be assumed to be high demand	36
Figure 6 – Examples of typical decomposition of a safety function into sub-functions and its allocation to subsystems	39
Figure 7 – Example of safety integrity of a safety function based on allocated subsystems as one SCS	40
Figure 8 – Subsystem A logical representation	60
Figure 9 – Subsystem B logical representation	60
Figure 10 – Subsystem C logical representation	60
Figure 11 – Subsystem D logical representation	61
Figure 12 – V-model for SW level 1.....	64
Figure 13 – V-model for software modules customized by the designer for SW level 1	64
Figure 14 – V-model of software safety lifecycle for SW level 2.....	70
Figure 15 – Overview of the validation process	79
Figure A.1 – Parameters used in risk estimation	92
Figure A.2 – Example proforma for SIL assignment process	98
Figure B.1 – Decomposition of the safety function.....	100
Figure B.2 – Overview of design of the subsystems of the SCS	100
Figure F.1 – Plant sketch	116
Figure F.2 – Principal module architecture design.....	119
Figure F.3 – Principal design approach of logical evaluation	120
Figure F.4 – Example of logical representation (program sketch)	121
Figure H.1 – Subsystem A logical representation	127
Figure H.2 – Subsystem B logical representation	128
Figure H.3 – Subsystem C logical representation.....	128
Figure H.4 – Correlation of subsystem C and the pertinent fault handling function	129
Figure H.5 – Subsystem C with external fault handling function	129
Figure H.6 – Subsystem C with external fault diagnostics	131
Figure H.7 – Subsystem C with external fault reaction	131
Figure H.8 – Subsystem C with internal fault diagnostics and internal fault reaction.....	131
Figure H.9 – Subsystem D logical representation.....	133
Figure I.1 – Example of a machine design plan including a safety plan	135
Figure I.2 – Example of activities, documents and roles	136
Table 1 – Terms used in IEC 62061	13
Table 2 – Abbreviations used in IEC 62061.....	28
Table 3 – SIL and limits of <i>PFH</i> values.....	36
Table 4 – Required SIL and <i>PFH</i> of pre-designed subsystem	40
Table 5 – Relevant information for each subsystem	47
Table 6 – Architectural constraints on a subsystem: maximum SIL that can be claimed for an SCS using the subsystem	56

Table 7 – Overview of basic requirements and interrelation to basic subsystem architectures	61
Table 8 – Different levels of application software	63
Table 9 – Documentation of an SCS	88
Table A.1 – Severity (Se) classification	93
Table A.2 – Frequency and duration of exposure (Fr) classification	94
Table A.3 – Probability (Pr) classification	95
Table A.4 – Probability of avoiding or limiting harm (Av) classification	96
Table A.5 – Parameters used to determine class of probability of harm (CI)	96
Table A.6 – Matrix assignment for determining the required SIL (or PL _r) for a safety function	97
Table B.1 – Safety requirements specification – example of overview	99
Table B.2 – Systematic integrity – example of overview	104
Table B.3 – Verification by tests	105
Table C.1 – Standards references and $MTTF_D$ or B_{10D} values for components	107
Table D.1 – Estimates for diagnostic coverage (DC)	109
Table E.1 – Criteria for estimation of CCF	112
Table E.2 – Criteria for estimation of CCF	113
Table F.1 – Example of relevant documents related to the simplified V-model	114
Table F.2 – Examples of coding guidelines	115
Table F.3 – Specified safety functions	117
Table F.4 – Relevant list of input and output signals	118
Table F.5 – Example of simplified cause and effect matrix	121
Table F.6 – Verification of software system design specification	122
Table F.7 – Software code review	122
Table F.8 – Software validation	123
Table G.1 – Examples of typical safety functions	124
Table H.1 – Allocation of PFH value of a subsystem	126
Table H.2 – Relationship between B_{10D} , operations and $MTTF_D$	127
Table H.3 – Minimum value of $1/\lambda_D F_H$ for the applicability of PFH equation (H.4)	132
Table J.1 – Minimum levels of independence for review, testing and verification activities	138
Table J.2 – Minimum levels of independence for validation activities	138

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) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

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

This second edition cancels and replaces the first edition, published in 2005, Amendment 1:2012 and Amendment 2:2015. This edition 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 text of this International Standard is based on the following documents:

Draft	Report on voting
44/885/FDIS	44/888/RVD

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

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 publication will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

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 63074.

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

Figure 1 illustrates the scope of this document.

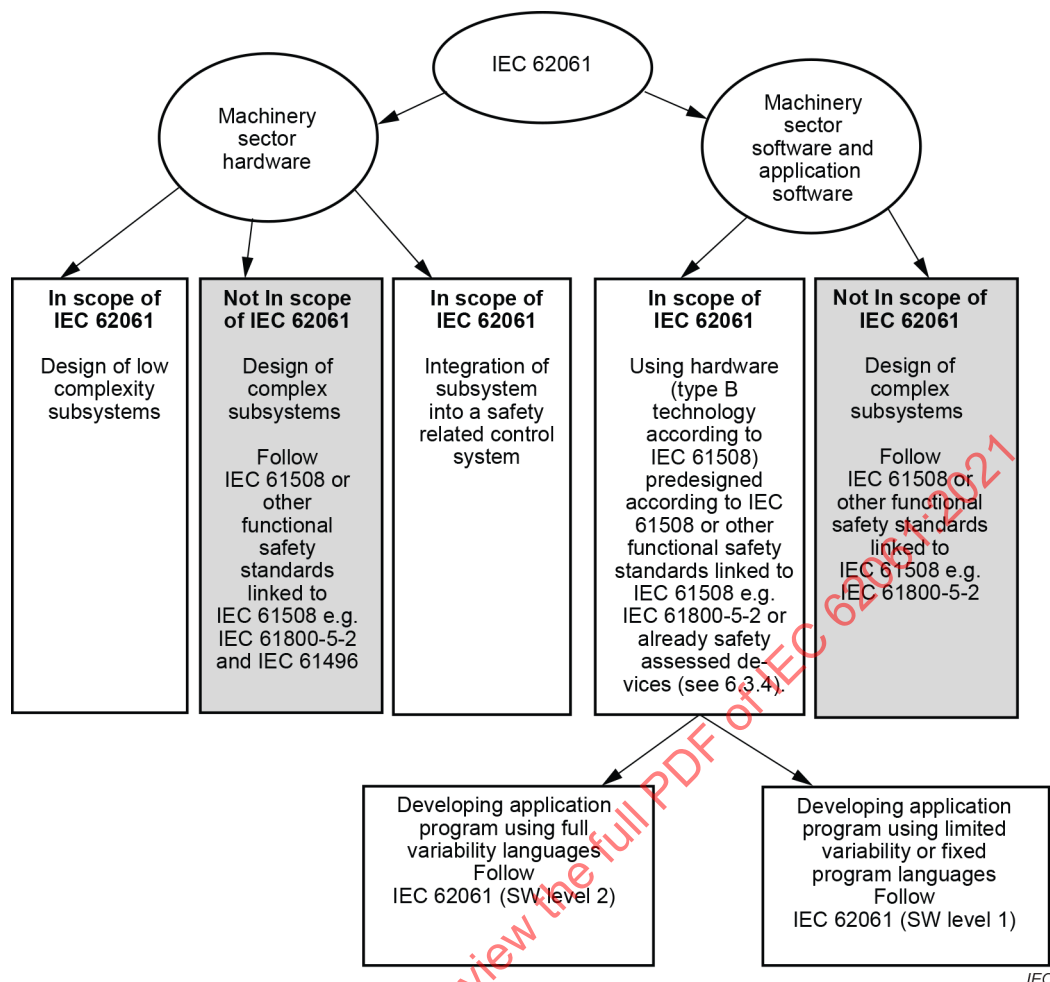


Figure 1 – Scope of this document

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 factors 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 $[t_1, t_2]$ is generally noted by $U(t_1, t_2)$.

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 $\lambda_{DU}(t)$ 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, 3.6.4, 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 suppressed 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.

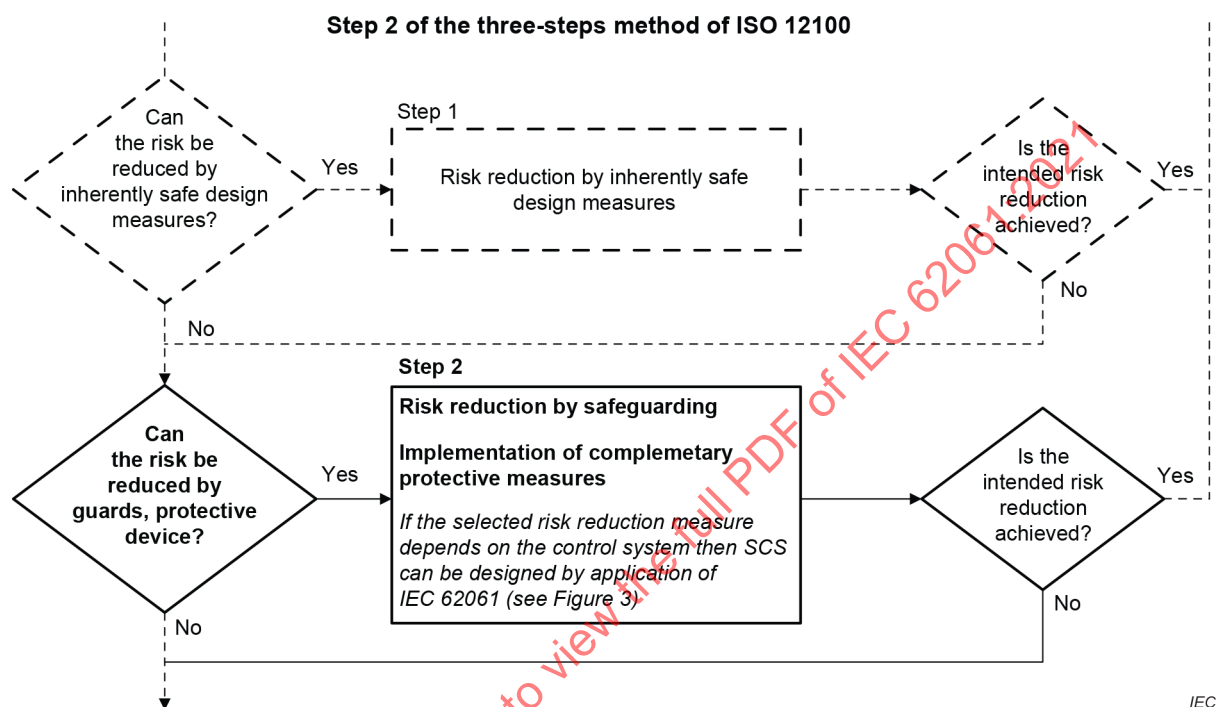


Figure 2 – Integration within the risk reduction process of ISO 12100 (extract)

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.

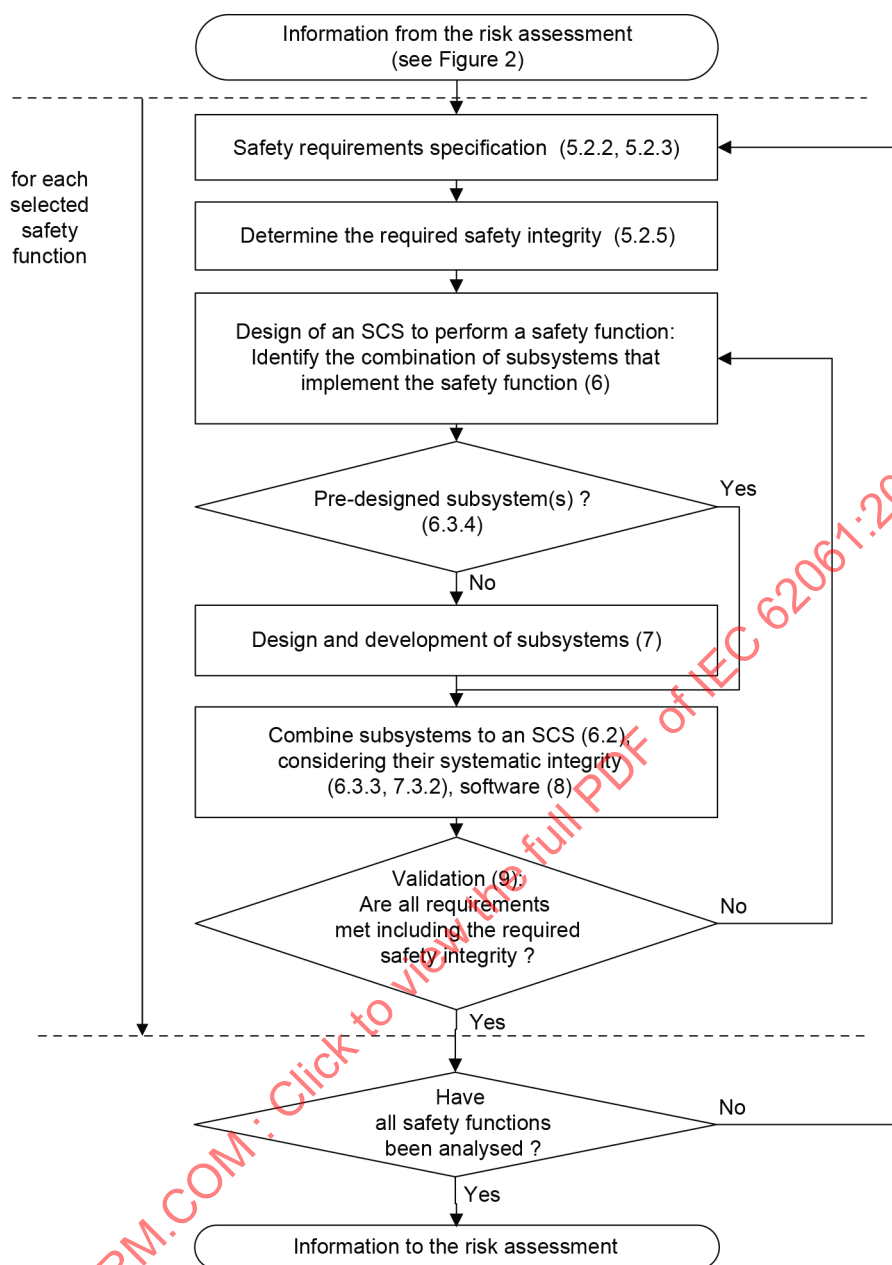


Figure 3 – Iterative process for design of the safety-related control system

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.

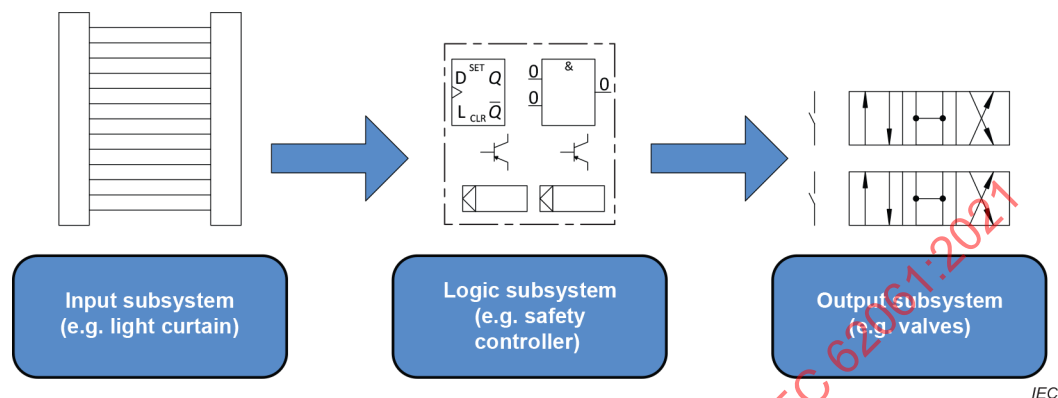


Figure 4 – Example of a combination of subsystems as one SCS

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.

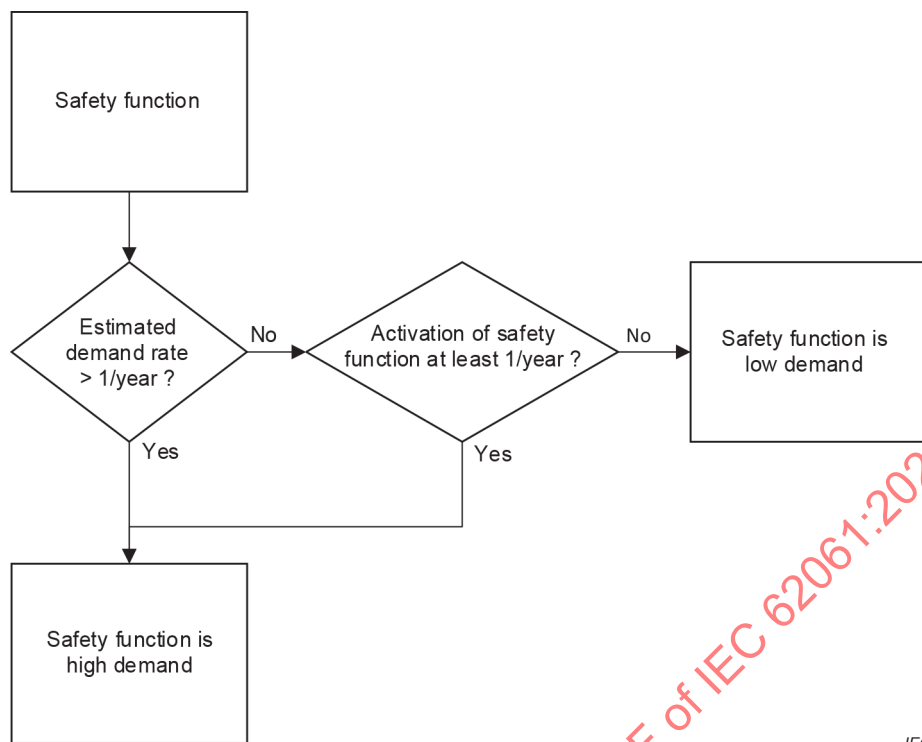
- rate of operating cycles, duty cycle, and/or utilisation category, for devices intended for use in the safety function;

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.



IEC

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 <i>PFH</i> 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).

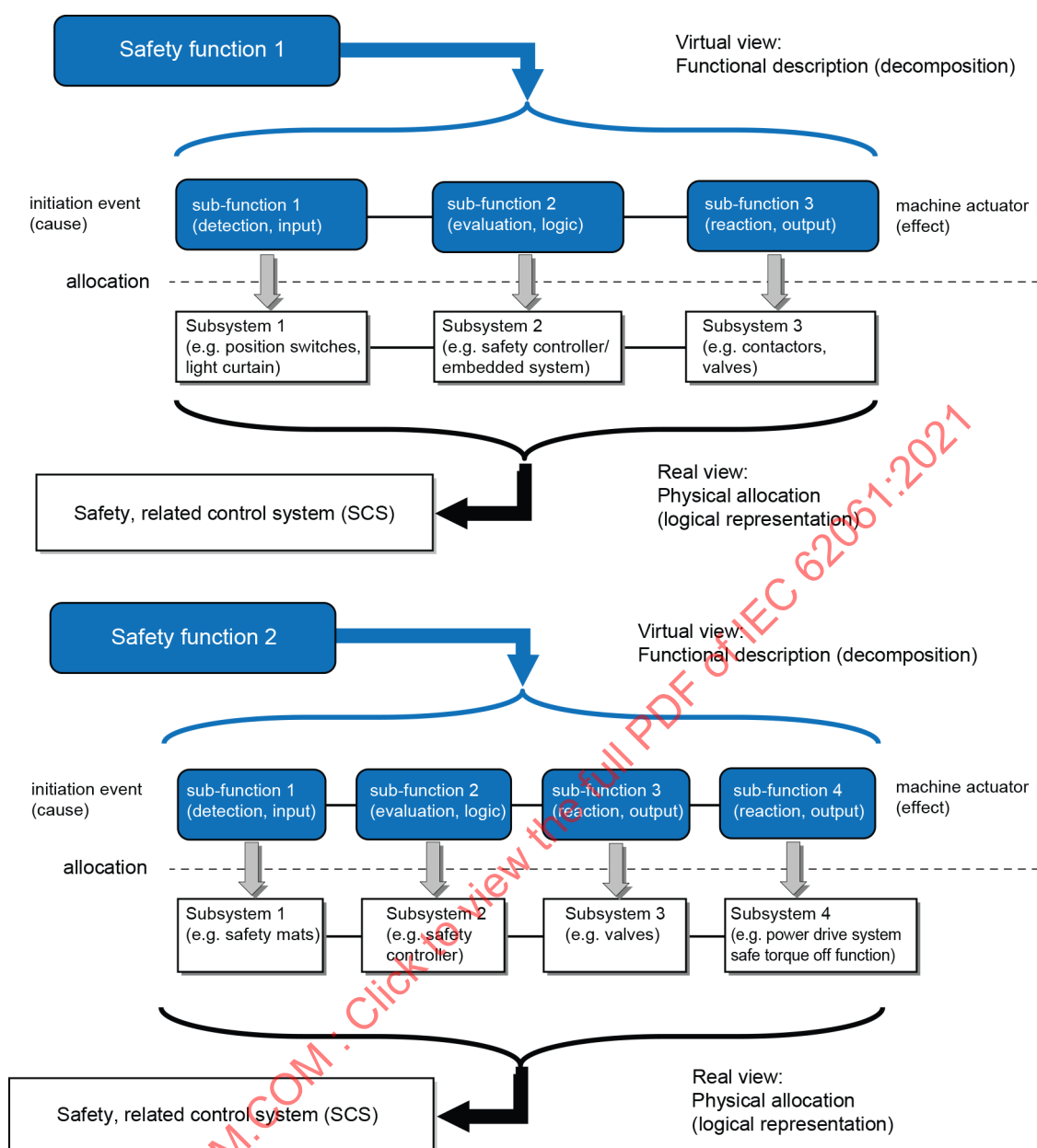
NOTE 4 A subsystem which implements a sub-function can consist of more than one physical unit. An example is a safety controller which has separate input, logic, output (and safety-related fieldbus communication) units. The manufacturer can provide separately the safety-related data for the units.

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.



IEC

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
<i>PFH</i>	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.

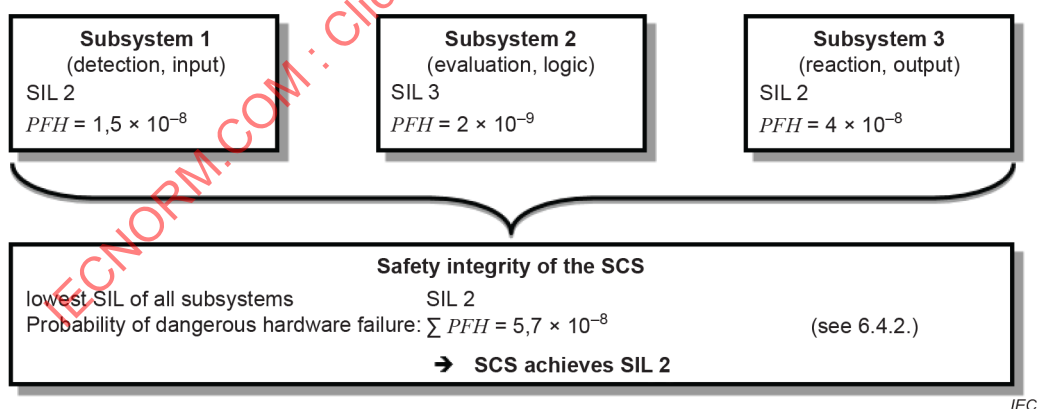


Figure 7 – Example of safety integrity of a safety function based on allocated subsystems as one SCS

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:

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

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, 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 63074, ISA TR84.00.09, ISO/IEC 27001:2013, 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 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-pre-designed) 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 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:

$$\lambda = \frac{1}{MTTF} \quad (3)$$

$$\lambda_D = \frac{1}{MTTF_D} \quad (4)$$

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.

$$1 FIT = 1 \times 10^{-9} h^{-1} \quad (5)$$

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

$$\lambda = \frac{1}{MTTF \times 8760 \frac{h}{a}} \quad (6)$$

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

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

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

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

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

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

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-trying 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

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

where

$$n_{op} = \frac{d_{op} \times h_{op} \times 3\,600 \frac{s}{h}}{t_{\text{cycle}}} \quad (8)$$

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 $1-e^{-\lambda t}$ 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} .

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 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
- 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, 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* ≥ 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:

- 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 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:

$$SFF = \frac{\sum \lambda_s + \sum \lambda_{DD}}{\sum \lambda_s + \sum \lambda_D} \quad (12)$$

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):

$$SFF = \frac{\sum \lambda_s + \sum \lambda_{DD}}{\sum \lambda_s + \sum \lambda_D} = \frac{\sum \lambda_{DD}}{\sum \lambda_D} \quad (13)$$

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

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

where DC_1 is the diagnostic coverage of subsystem element 1.

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

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

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:

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

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.

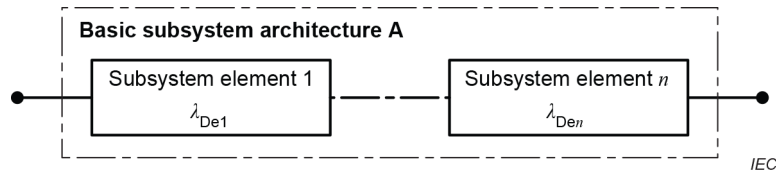


Figure 8 – Subsystem A logical representation

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.

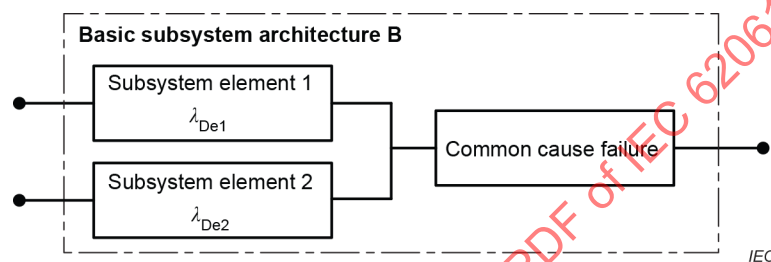


Figure 9 – Subsystem B logical representation

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.

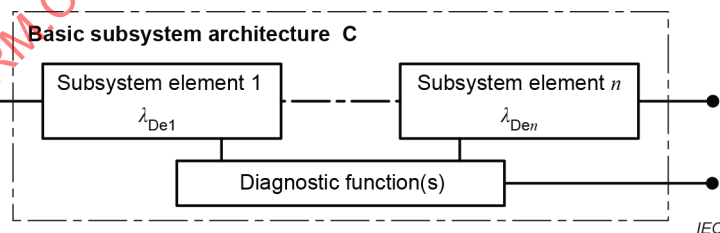


Figure 10 – Subsystem C logical representation

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.

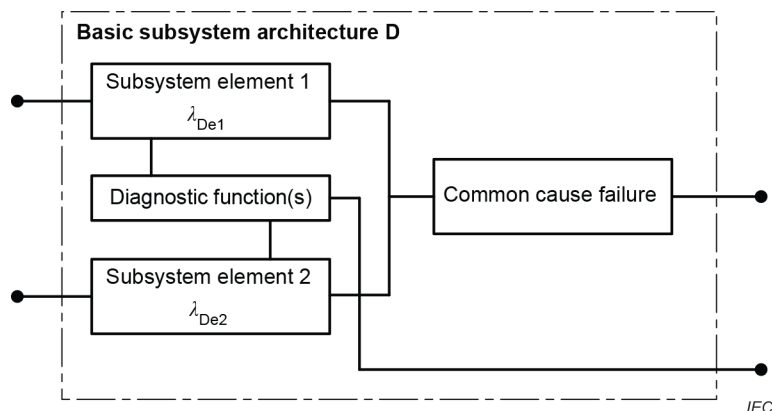


Figure 11 – Subsystem D logical representation

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 complying with this document.	Safety PLC with FVL (FVL complying with this document.)
3	Application software making use of another language than a limited variability language (LVL).	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.

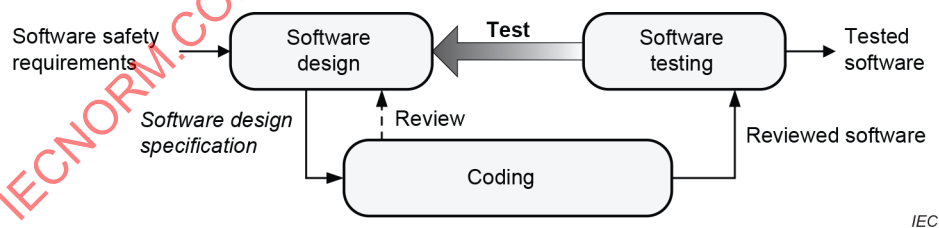


Figure 12 – V-model for SW level 1

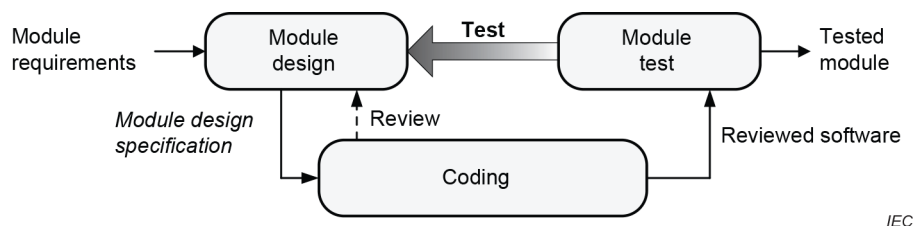


Figure 13 – V-model for software modules customized by the designer for SW level 1

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.

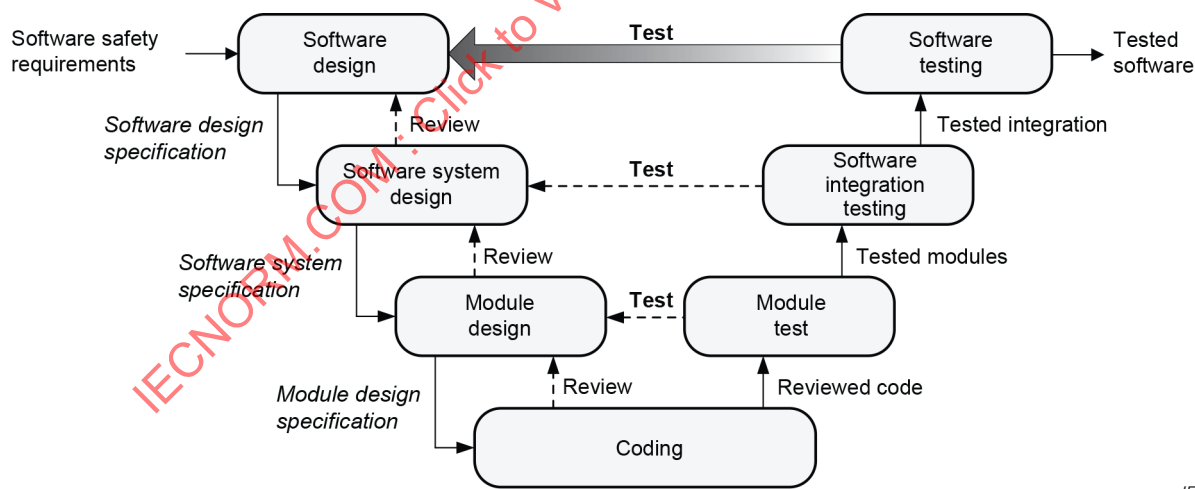
Any software lifecycle model may be used provided all the objectives and requirements of this Subclause 8.4 are met. Safety-related software shall be validated as described in 9.5.3.

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.



IEC

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.

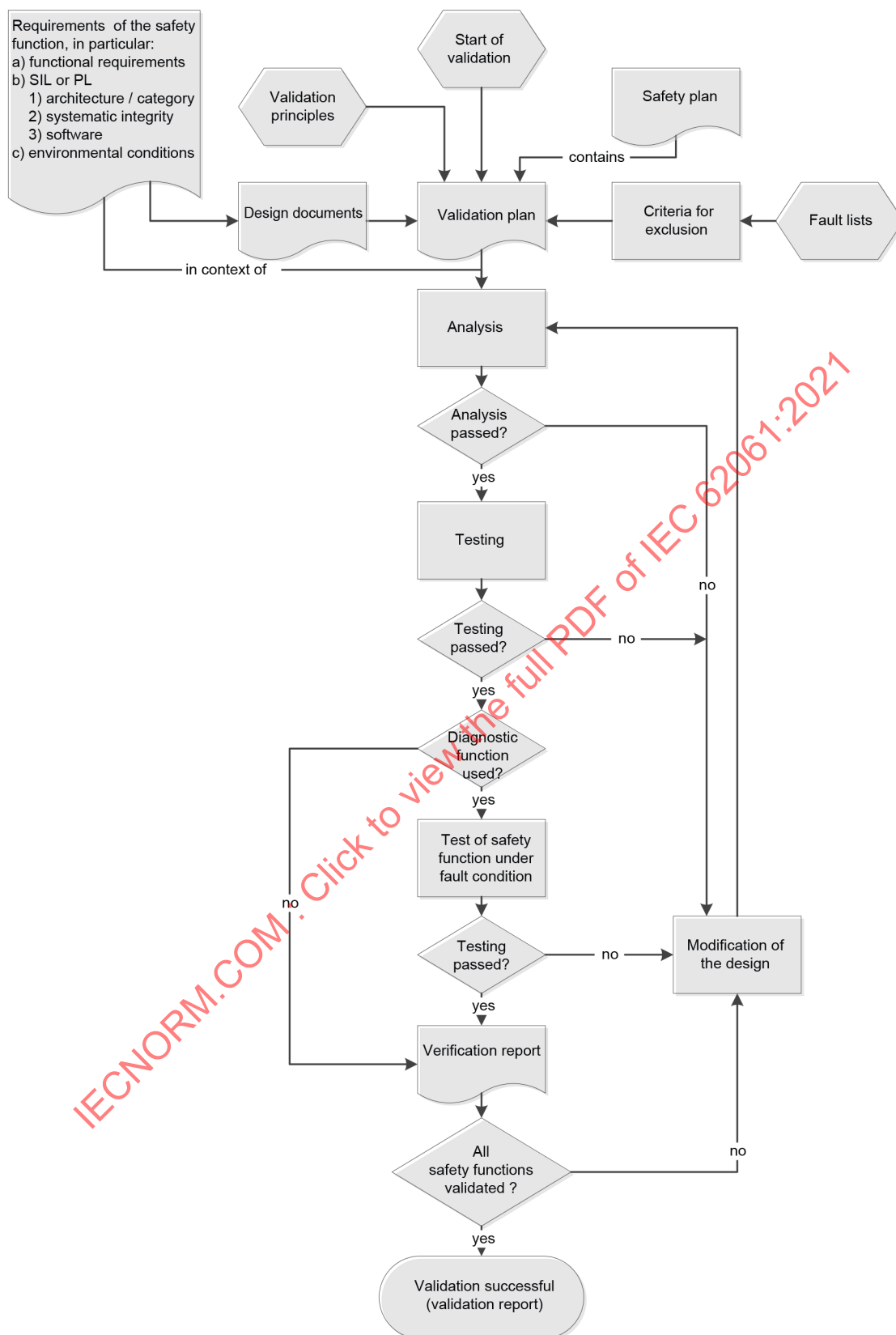


Figure 15 – Overview of the validation process

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, hydrolysis, 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: $SIL \geq \text{required SIL}$.

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".

IECNORM.COM : Click to view the full PDF of IEC 62061:2021

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

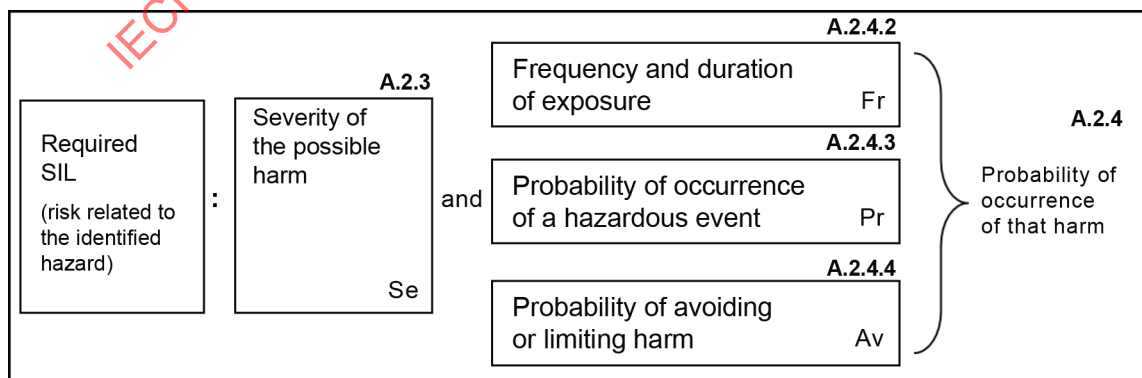
A.2.1 Hazard identification/indication

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 .



IEC

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 $\geq$ 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 $\geq$ 10 min	Duration of exposure < 10 min
$\geq$ 1 per h	5	5
< 1 per h to $\geq$ 1 per day	5	4
< 1 per day to $\geq$ 1 per 2 weeks	4	3
< 1 per 2 weeks to $\geq$ 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 (CI)

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 CI (where $CI = Fr + Pr + Av$) in Table A.5.

Table A.5 – Parameters used to determine class of probability of harm (CI)

ID.	Hazard	Fr	Pr	Av	CI
1					
2					
3					
4					

A.2.6 SIL assignment

Using Table A.6, where the severity (Se) row crosses the relevant column (CI), 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

Consequences	Severity Se	Class CI = Fr + Pr + Av															
		3	4	5	6	7	8	9	10	11	12	13	14	15			
Death, losing an eye or arm	4	SIL 1		SIL 2			SIL 2			SIL 3			SIL 3				
		PL _r b	PL _r c	PL _r d			PL _r d			PL _r e			PL _r e				
Permanent injury, losing fingers	3	No SIL (or PL) required		OM			SIL 1			SIL 2			SIL 3				
				PL _r a			PL _r b	PL _r c	PL _r d			PL _r e					
Reversible injury, medical attention	2			OM			SIL 1			SIL 2							
							PL _r a			PL _r b	PL _r c	PL _r d					
Reversible injury, first aid	1			OM			SIL 1			SIL 1							
							PL _r a			PL _r b	PL _r c	PL _r d					
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: CI = 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.																	
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 OM: Other Measures (e.g. basic safety principles, Table 7)																	

Figure A.2 shows an example of documentation.

[illegible]

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	
<i>Description:</i>	When the guard door will be opened then the electrical motor shall stop
<i>Condition(s):</i>	Operating mode “all”
<i>Restart/Reset</i>	Manual human action required where the operator can stay in the hazard zone (according to ISO 12100:2010, 6.2.11.3)
<i>Priority:</i>	High priority in comparison to other safety functions; emergency stop function will have the highest priority
<i>Frequency of operation:</i>	10 time per hour; 16 hours per day; 250 days per year
<i>Response time:</i>	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)
<i>Interface(s) to other machine functions:</i>	Realized by a pre-designed safety-related device (information for use of the component manufacturer to be referenced)
<i>Fault reaction function:</i>	Stopping immediately or detection at restart at least by prohibiting restart
<i>Defeating:</i>	Design of guard door and mounting of guard interlocking devices according to ISO 14119
<i>Environment</i>	Temperature, dust, vibration, ...
Safety integrity requirements specification of the safety function	
<i>Required SIL or PL</i>	SIL 2 with related target <i>PFH</i> value (see Table 1)
<i>Architectural constraints</i>	– 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.

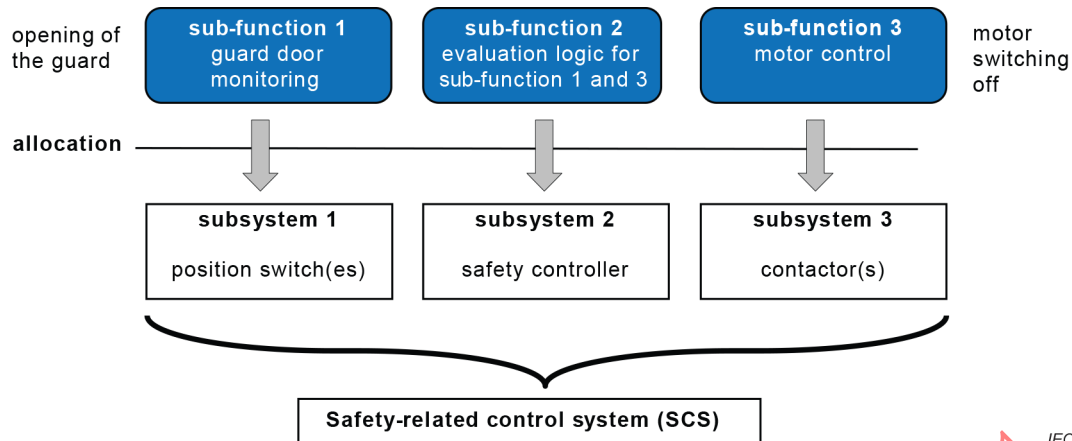


Figure B.1 – Decomposition of the safety function

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.

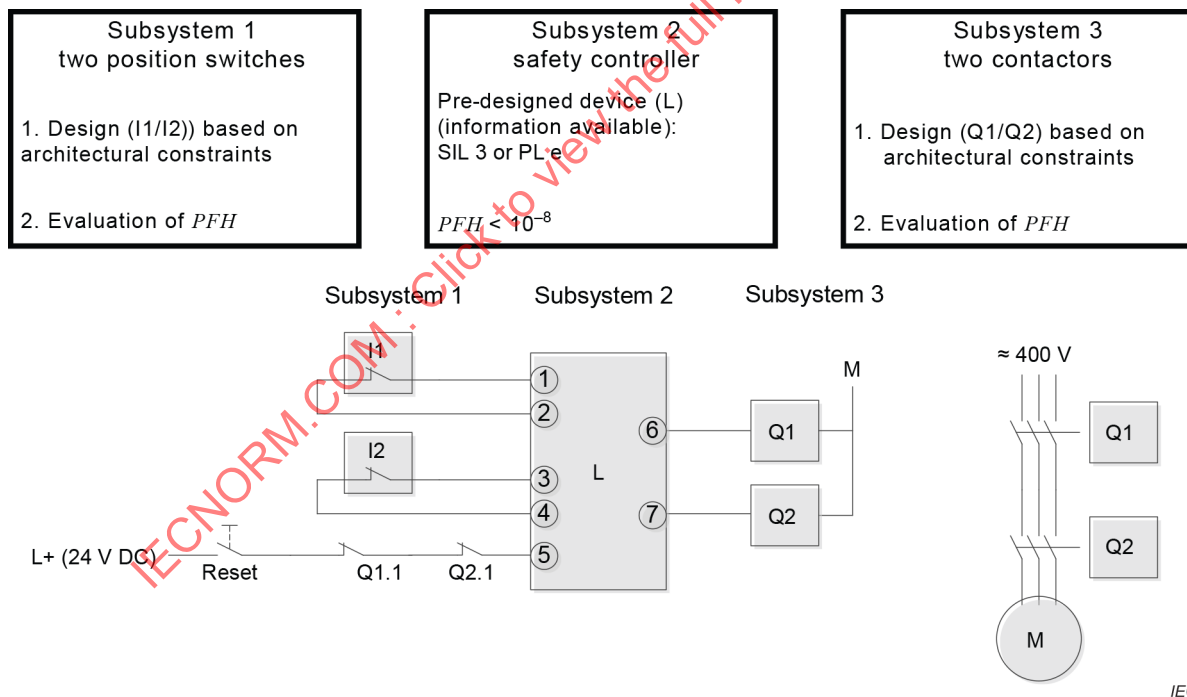


Figure B.2 – Overview of design of the subsystems of the SCS

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 $\lambda_S = 0$.

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

- $\lambda_S = 0$ 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):

$$\begin{aligned}
 - n_{op} &= \frac{d_{op} \times h_{op} \times 3\,600 \frac{s}{h}}{t_{cycle}} = \frac{250 d \times 16 h / d \times 3\,600 \frac{s}{h}}{360 s} = 40\,000 \text{ cycles per year;} \\
 - MTTF_D &= \frac{B_{10D}}{0,1 \times n_{op}} = \frac{2\,000\,000}{0,1 \times 40\,000} = 500 \text{ years;} \\
 - \lambda_D &= \frac{1}{MTTF_D \times 8\,760 h/a} = 2,28 \times 10^{-7} \text{ failures per hour.}
 \end{aligned}$$

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): $PFH = 0,10 \times 10^{-7}$. This approach is simple but gives a conservative approach.
- Formulas (see H.2.5): $PFH = 4,65 \times 10^{-9}$ (with $\beta = 2\%$, $T_1 = 175\,200$ h, $T_2 = 1/C = n_{op}/8\,760$ h). 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):

$$\begin{aligned}
 - \quad n_{op} &= \frac{d_{op} \times h_{op} \times 3\,600 \frac{s}{h}}{t_{cycle}} = \frac{250 \frac{d}{h} \times 16 \frac{h}{d} \times 3\,600 \frac{s}{h}}{360 \text{ s}} = 40\,000 \text{ cycles per year ;} \\
 - \quad MTTF_D &= \frac{B_{10D}}{0,1 \times n_{op}} = \frac{1\,300\,000}{0,1 \times 40\,000} = 325 \text{ years ;} \\
 - \quad \lambda_D &= \frac{1}{MTTF_D \times 8\,760 \frac{h}{a}} = 3,51 \times 10^{-7} \text{ failures per hour .}
 \end{aligned}$$

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): $PFH = 0,10 \times 10^{-7}$. This approach is simple but gives a conservative result.
- Formulas (see H.2.5): $PFH = 7,23 \times 10^{-9}$ (with $\beta = 2 \%$, $T_1 = 175\,200 \text{ h}$, $T_2 = 1/C = n_{op}/8\,760 \text{ h}$). 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).

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).

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:

- 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).
- The manufacturer of the component specifies the appropriate application and operating conditions for the subsystem designer.
- 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:

- The manufacturer of the hydraulic component specifies the appropriate application and operating conditions for the subsystem designer, and
- 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 $n_{op} \geq 1\,000\,000$	Tables C.1 and C.2	ISO 4413	$MTTF_D = 150$
Hydraulic components $1\,000\,000 > n_{op} \geq 500\,000$	Tables C.1 and C.2	ISO 4413	$MTTF_D = 300$
Hydraulic components $500\,000 > n_{op} \geq 250\,000$	Tables C.1 and C.2	ISO 4413	$MTTF_D = 600$
Hydraulic components $250\,000 > n_{op}$	Tables C.1 and C.2		$MTTF_D = 1\,200$
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 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 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.

IECNORM.COM : Click to view the full PDF of IEC 62061:2021

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 (<i>DC</i>)	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 → <i>DC</i> = 90 % Emergency stop dual channel with short circuit detection → <i>DC</i> = 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 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 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 simple qualitative approaches 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 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

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)

IECNORM.COM : Click to view the full PDF of IEC 62061:2021

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

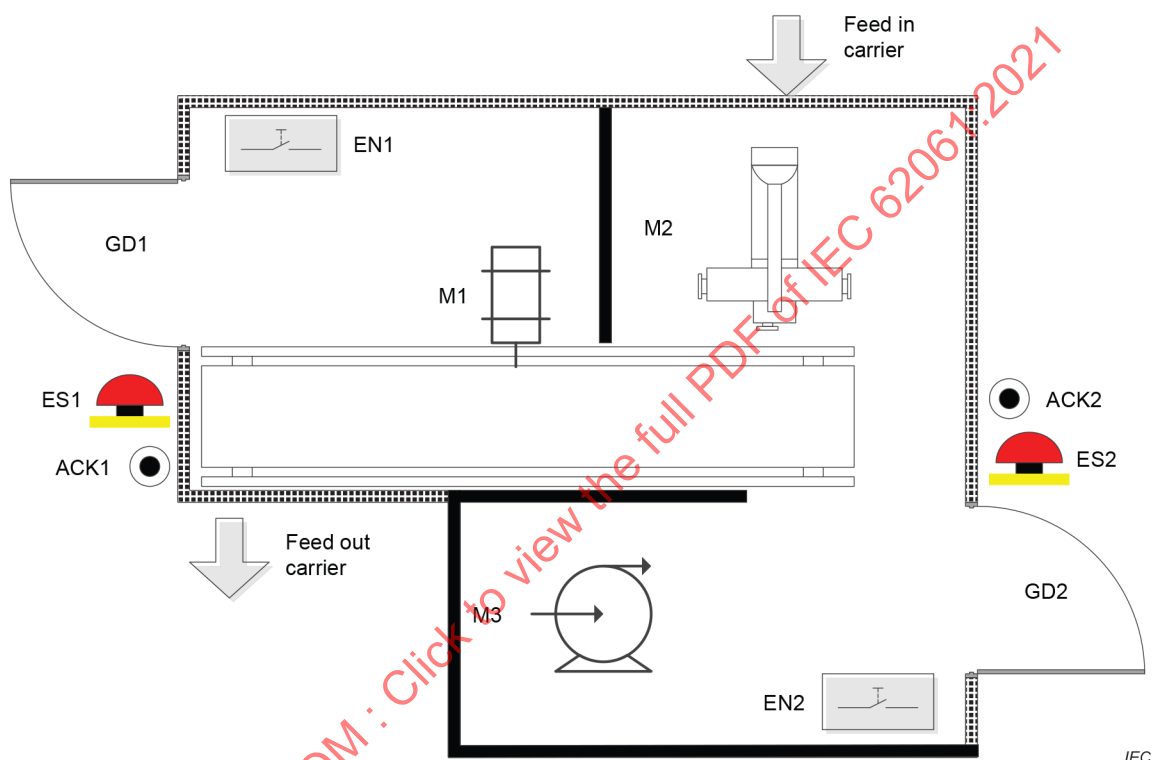
A. Variables
Prefixes of boolean variables: "b".
Prefixes of binary inputs: "I_b" (non-safety-related input), "IS_b" (safety-related input).
Prefixes of binary outputs: "Q_b" (non-safety-related output) or "QS_b" (safety-related input).
Prefixes of instances: Timers: "T_", positive edge detections: "R_", 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.
Contactors: 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.

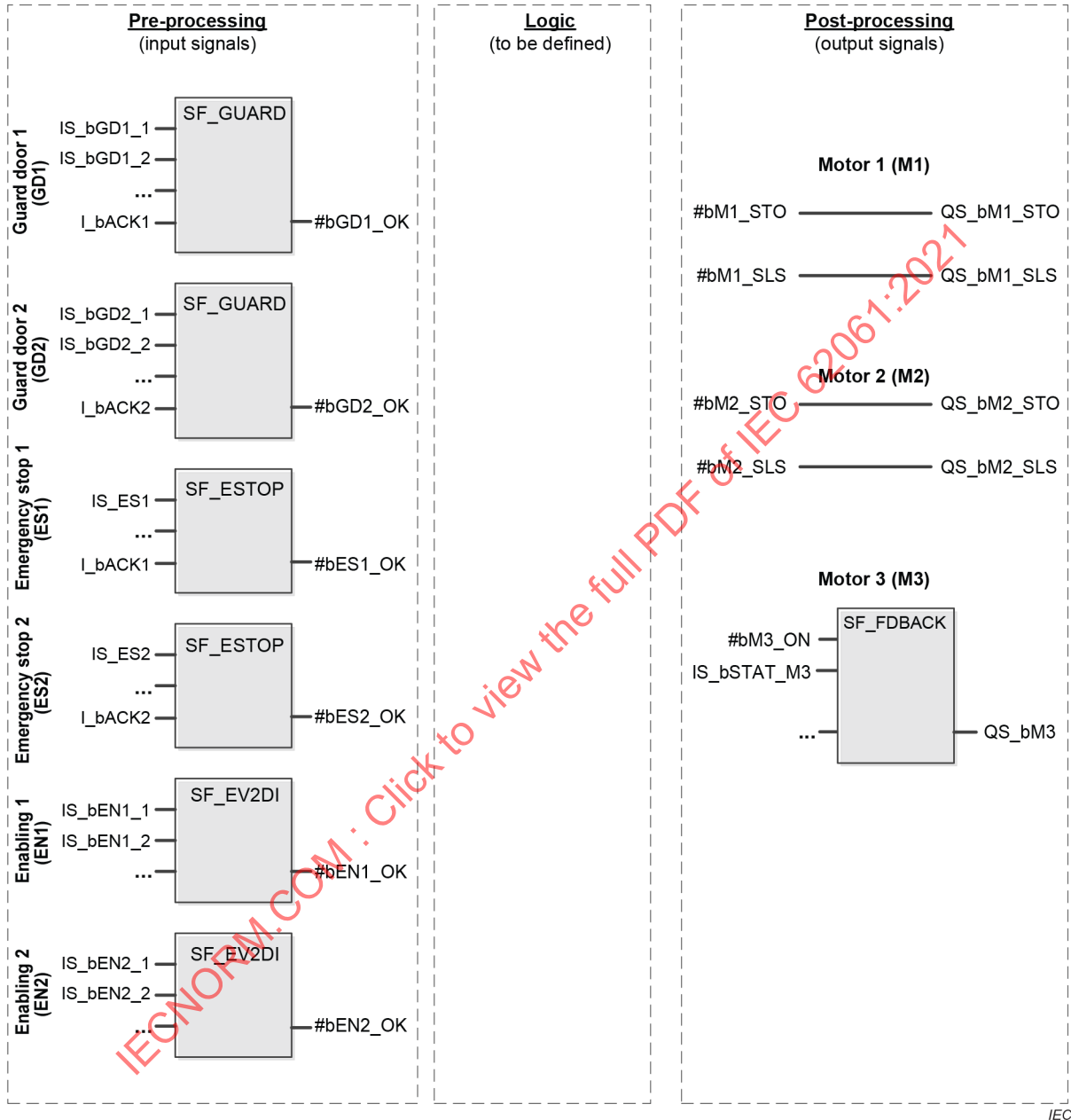


Figure F.2 – Principal module architecture design

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.

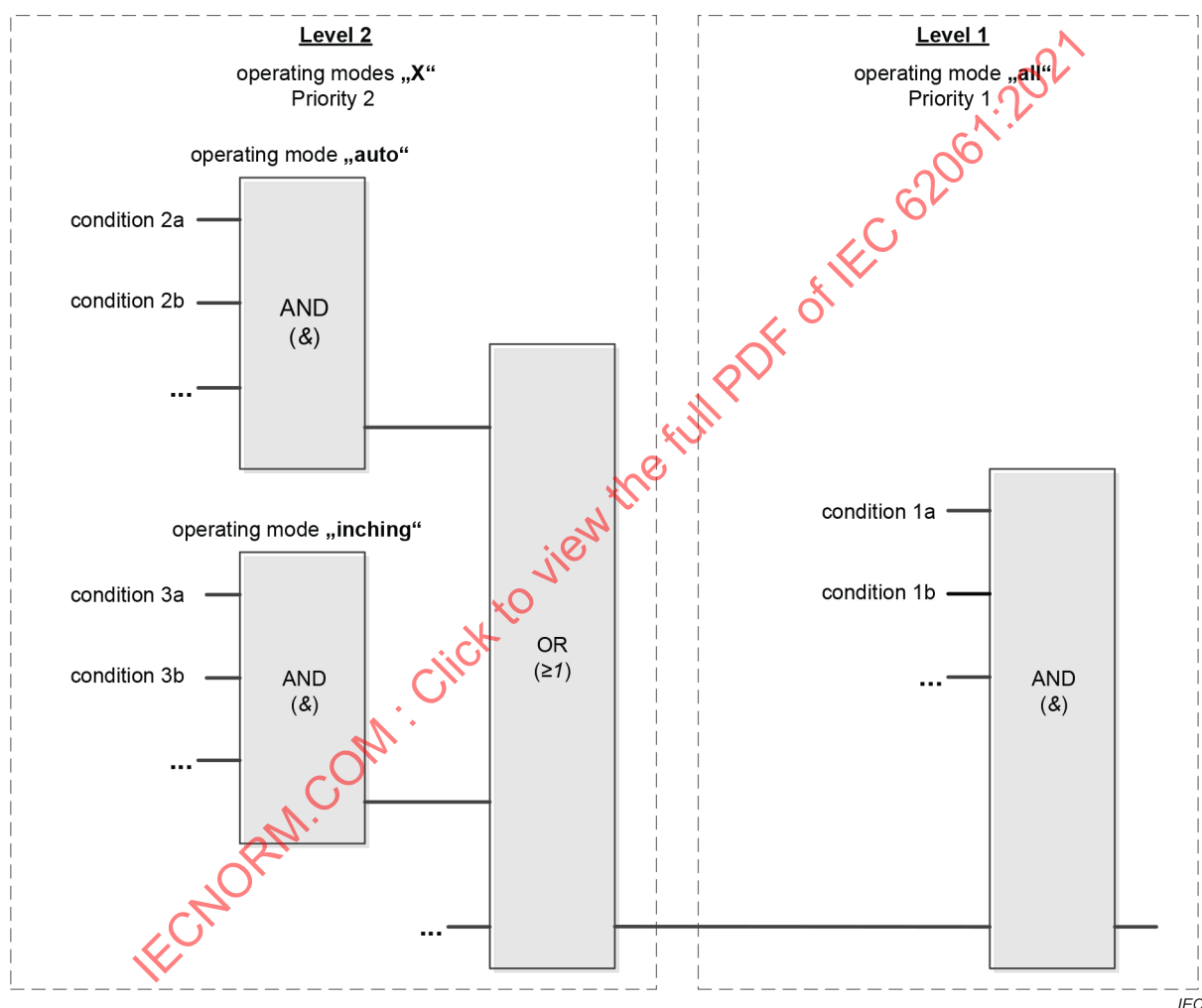


Figure F.3 – Principal design approach of logical evaluation

Figure F.4 represents logic evaluation of the safety functions described in Table F.3.

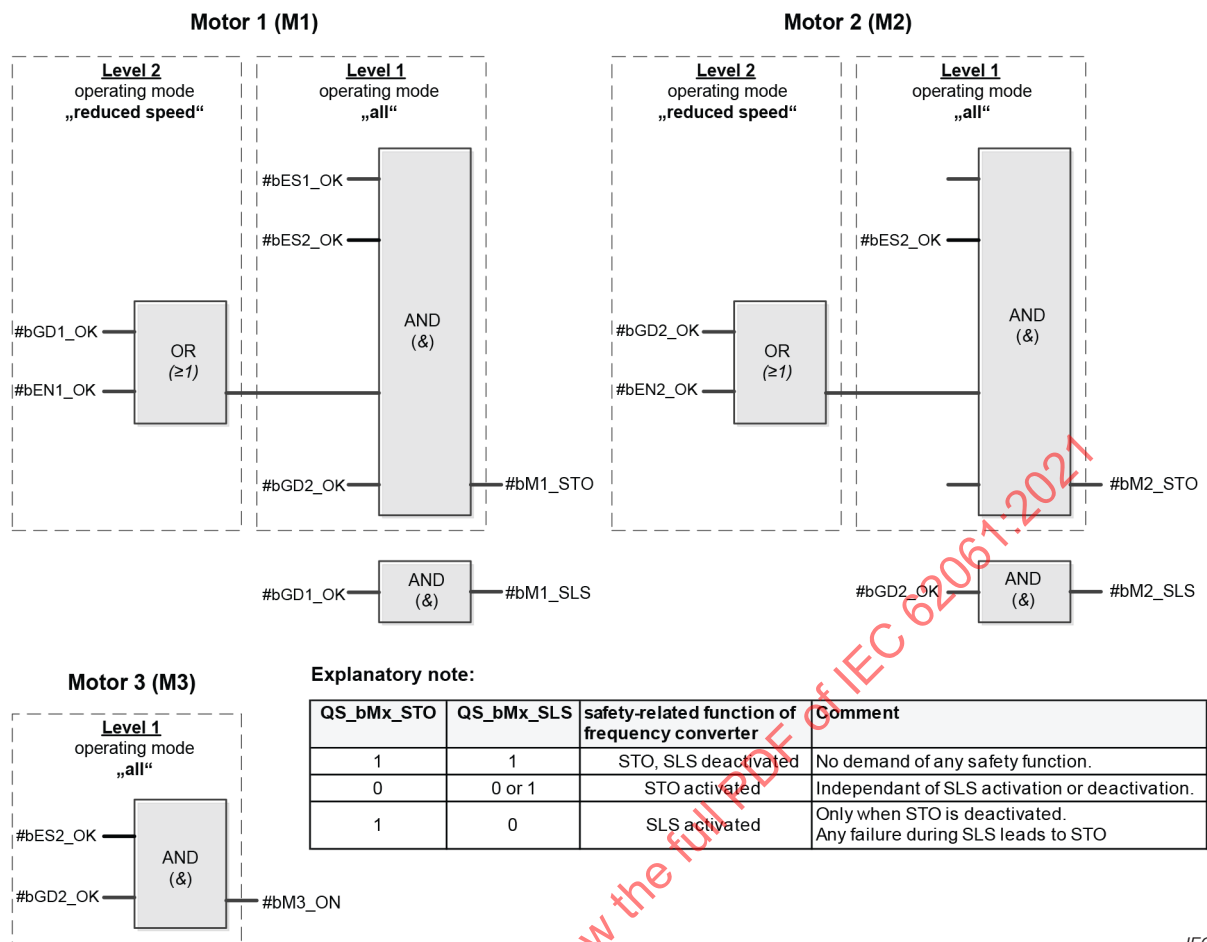


Figure F.4 – Example of logical representation (program sketch)

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, converters, 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) – device with reset (push button) – two-hand control – hold-to-run devices	6.2.11.8 6.2.11.8 b) 6.2.11.8 b) 6.2.11.9 6.2.11.8 b)	IEC 60204-1, Type C standards ISO 11161 Integrated manufacturing systems Type C standards ISO 13851, Two-hand control devices
Adjusting, teaching, retooling, fault finding, maintenance, cleaning – enabling function – safe motion, safe “positioning”		IEC 60204-1, Type C standards IEC 60947-5-8 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: $MTTF_D = \sqrt{20 \text{ a} \times 200 \text{ a}} = 63,2$ 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, $0,1 \times 10^{-06} = 1 \times 10^{-07}$ as 10 % of the *PFH* value for SIL 2 can be allocated;
- with $DC_1 = 90$ %, $DC_2 \geq 99$ % and $MTTF_D = 50$ years per channel, $0,2 \times 10^{-06} = 2 \times 10^{-07}$ as 20 % of the *PFH* value for SIL 2 with $DC \geq 90$ % can be allocated;
- with $DC_1 = 90$ %, $DC_2 = 90$ % and $MTTF_D = 20$ years per channel, $0,3 \times 10^{-05} = 3 \times 10^{-06}$ as 30 % of the *PFH* value for SIL 1 with $DC \geq 60$ % 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 $0,5 \times 10^{-07} = 5 \times 10^{-08}$ as 30 % 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												PFH [h ⁻¹]					
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								
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					
		60 – < 69	90 %			23 – < 26	90 %	9 – < 11	99 %	→	50 %	5 × 10 ⁻⁷	SIL 2	10 ⁻⁶			
		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					
								54 – < 65	99 %	→	50 %	5 × 10 ⁻⁸	SIL 3	10 ⁻⁷			
								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					
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 factor of $\beta = 2\%$ 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			MTTF _D per channel [years] based on operations and B_{10D}								
cycle interval	duty cycle (per hour)	annual operations n_{op}	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
30 sec	120	1.051.200	not recommended		10	12	19	48	95	190	476
1 min	60	525.600			19	25	38	95	190	381	951
2 min	30	262.800		15	38	49	76	190	381	761	1903
5 min	12	105.120	10	38	95	124	190	476	951	1903	
10 min	6	52.560	19	76	190	247	381	951	1903		
15 min	4	35.040	29	114	285	371	571	1427			
30 min	2	17.520	57	228	571	742	1142				
1 h	1	8.760	114	457	1142	1484					
2 h	0,5	4.380	228	913	2283						
>= 4 h	0,25	2.190	457	1826							

MTTF_D > 2000 years

interval or operations based on:
24 hours per day
365 days per year
(1 year = 8760 h)

$T_{10D} \approx MTTF_D / 10$

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	position switches
	enabling switches						pushbuttons	pushbuttons	pushbuttons	
actuators		proximity switches with nominal load						proximity switches with small load		
	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

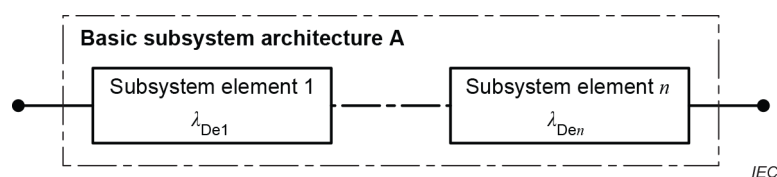
H.2.1 General

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 ($1 \gg \lambda \times T_1$) 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 – Subsystem A logical representation**

For architecture A, the PFH of the subsystem is the sum of the dangerous failure rates of all subsystems elements:

$$PFH = \lambda_{De1} + \dots + \lambda_{De_n} \quad (H.1)$$

where

λ_{Dei} is the dangerous failure rate of element e_i 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.

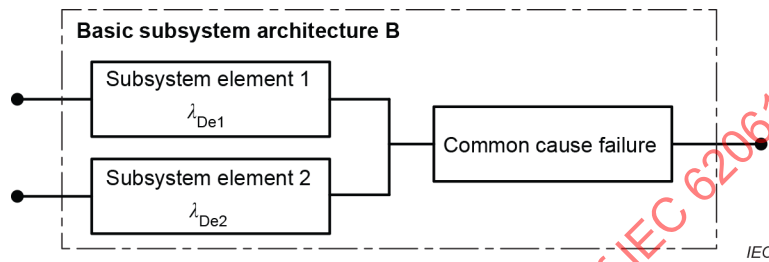


Figure H.2 – Subsystem B logical representation

For architecture B, the PFH of the subsystem is:

$$PFH = (1 - \beta)^2 \times \lambda_{De1} \times \lambda_{De2} \times T_1 + \beta \times (\lambda_{De1} + \lambda_{De2}) / 2 \quad (H.2)$$

where

λ_{De1} is the dangerous failure rate of element e_1 comprising the first functional channel;

λ_{De2} is the dangerous failure rate of element e_2 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).

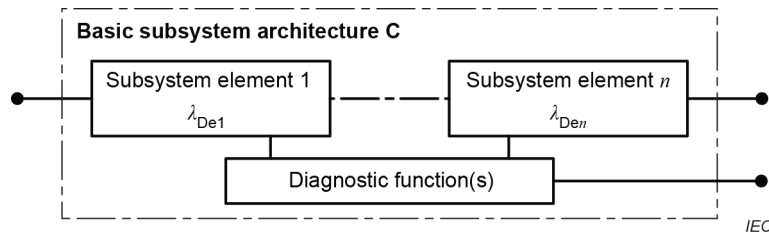


Figure H.3 – Subsystem C logical representation

The safety function is performed by a single channel comprising the elements e_1 to e_n . 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).

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.

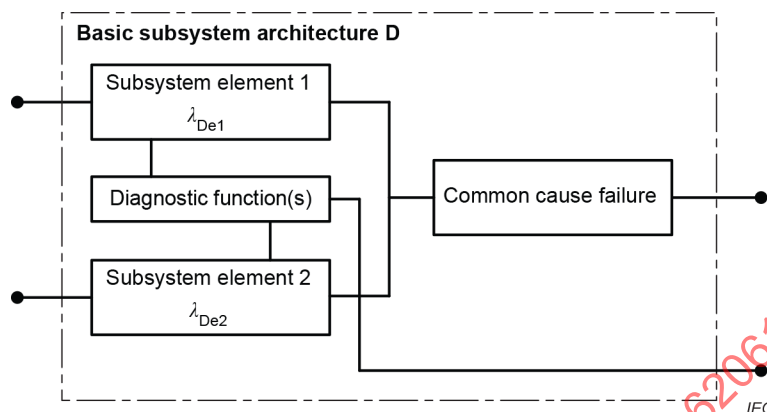


Figure H.4 – Correlation of subsystem C and the pertinent fault handling function

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.

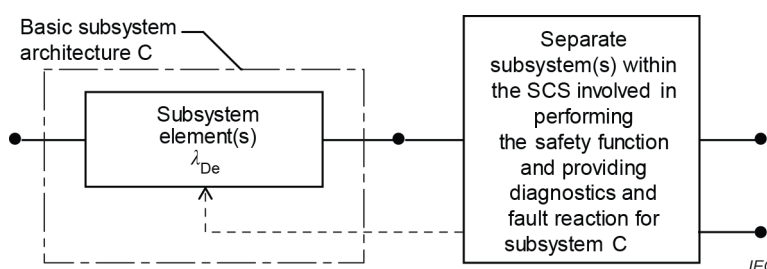


Figure H.5 – Subsystem C with external fault handling function

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:

$$PFH = (1 - DC_1) \times \lambda_{De1} + \dots + (1 - DC_n) \times \lambda_{Den} \quad (H.3)$$

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 n^{th} 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 n^{th} 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.

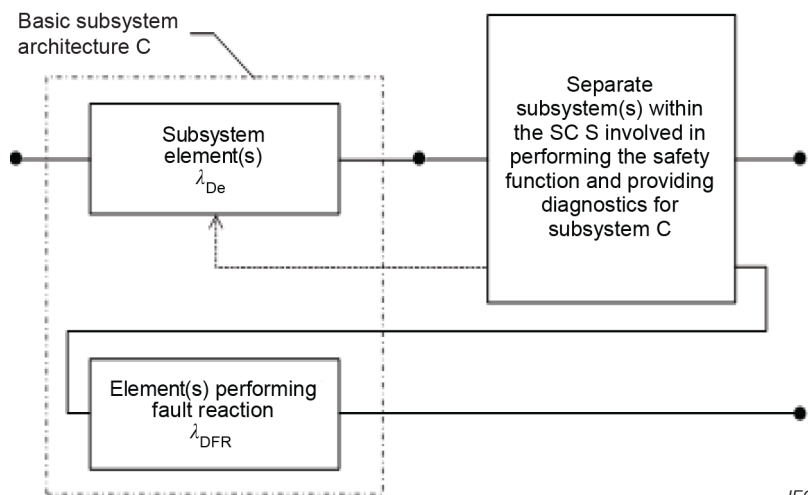


Figure H.6 – Subsystem C with external fault diagnostics

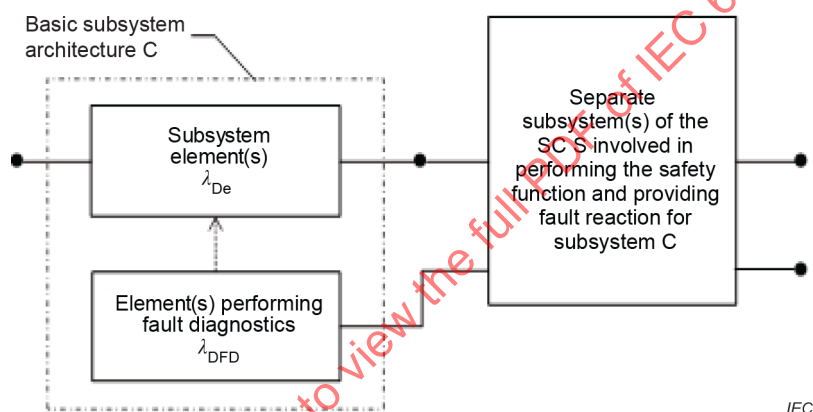


Figure H.7 – Subsystem C with external fault reaction

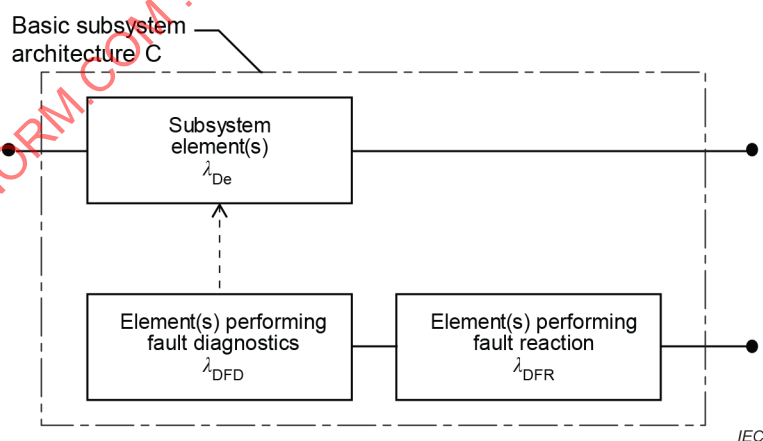


Figure H.8 – Subsystem C with internal fault diagnostics and internal fault reaction

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;

- $1/\lambda_{D\text{ FH}}$ has at least the minimum value according to Table H.3;

where $\lambda_{D\text{ FH}}$ 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.4).

NOTE 1 Common cause failures factor (β) is considered between the channel (λ_{De}) and the fault reaction function channel (λ_{DFR}).

Table H.3 – Minimum value of $1/\lambda_{D\text{ FH}}$ for the applicability of *PFH* equation (H.4)

<i>DC</i> range	Minimum value of $1/\lambda_{D\text{ FH}}$ (years)
$60\% \leq DC < 65\%$	44
$65\% \leq DC < 70\%$	59
$70\% \leq DC < 75\%$	100
$75\% \leq DC < 80\%$	170
$80\% \leq DC < 85\%$	300
$85\% \leq DC < 90\%$	550
$90\% \leq DC < 95\%$	1 200
$95\% \leq DC \leq 99\%$	5 900
NOTE If the functional channel is comprised by more than one element, the related <i>DC</i> 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*:

$$PFH = \lambda_{De} - DC \times \left[\lambda_{De} - \beta \times \min(\lambda_{De}, \lambda_{D\text{ FH}}) \right] \times \left\{ 1 - \frac{1}{2} \left[\lambda_{D\text{ FH}} - \beta \times \min(\lambda_{De}, \lambda_{D\text{ FH}}) \right] \times T_1 \right\} \quad (\text{H.4})$$

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;
- $\lambda_{D\text{ FH}}$ 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*:

$$PFH = \sum_{i=1}^n \lambda_{Dei} - DC \times \left(\sum_{i=1}^n \lambda_{Dei} - \lambda_{CC} \right) \times \left\{ 1 - \frac{1}{2} \left[\sum_{j=1}^m \lambda_{D FH j} - \lambda_{CC} \right] \times T_1 \right\} \quad (H.5)$$

with

$$DC = \frac{\sum_{i=1}^n (DC_i \times \lambda_{Dei})}{\sum_{i=1}^n \lambda_{Dei}} \quad (H.6)$$

$$\lambda_{CC} = \beta \times \min \left(\sum_{i=1}^n \lambda_{Dei}, \sum_{j=1}^m \lambda_{D FH j} \right) \quad (H.7)$$

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)

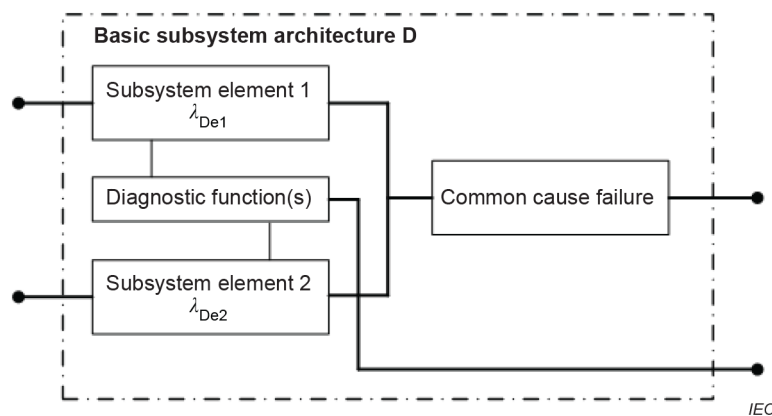


Figure H.9 – Subsystem D logical representation

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:

$$PFH = (1 - \beta)^2 \times [\lambda_{De1} \times \lambda_{De2} \times (DC_1 + DC_2) \times T_2 / 2 + \lambda_{De1} \times \lambda_{De2} \times (2 - DC_1 - DC_2) \times T_1 / 2] + \beta \times (\lambda_{De1} + \lambda_{De2}) / 2 \quad (H.8)$$

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:

$$PFH = (1 - \beta)^2 \times [DC \times T_2 + (1 - DC) \times T_1] \times \lambda_{De}^2 + \beta \times \lambda_{De} \quad (H.9)$$

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.

$$\lambda_D = \sum_{i=1}^N \lambda_{Di} = \sum_{j=1}^{\tilde{N}} (n_j \lambda_{Dj}) \quad (H.10)$$

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)

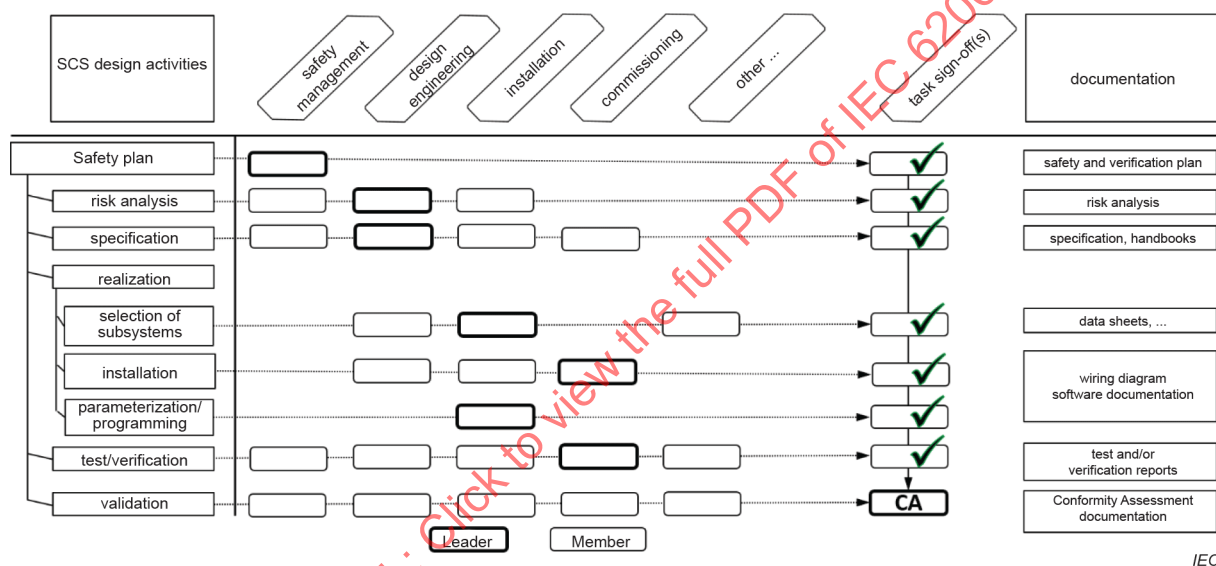
The functional safety plan and design activities

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.



IEC

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.

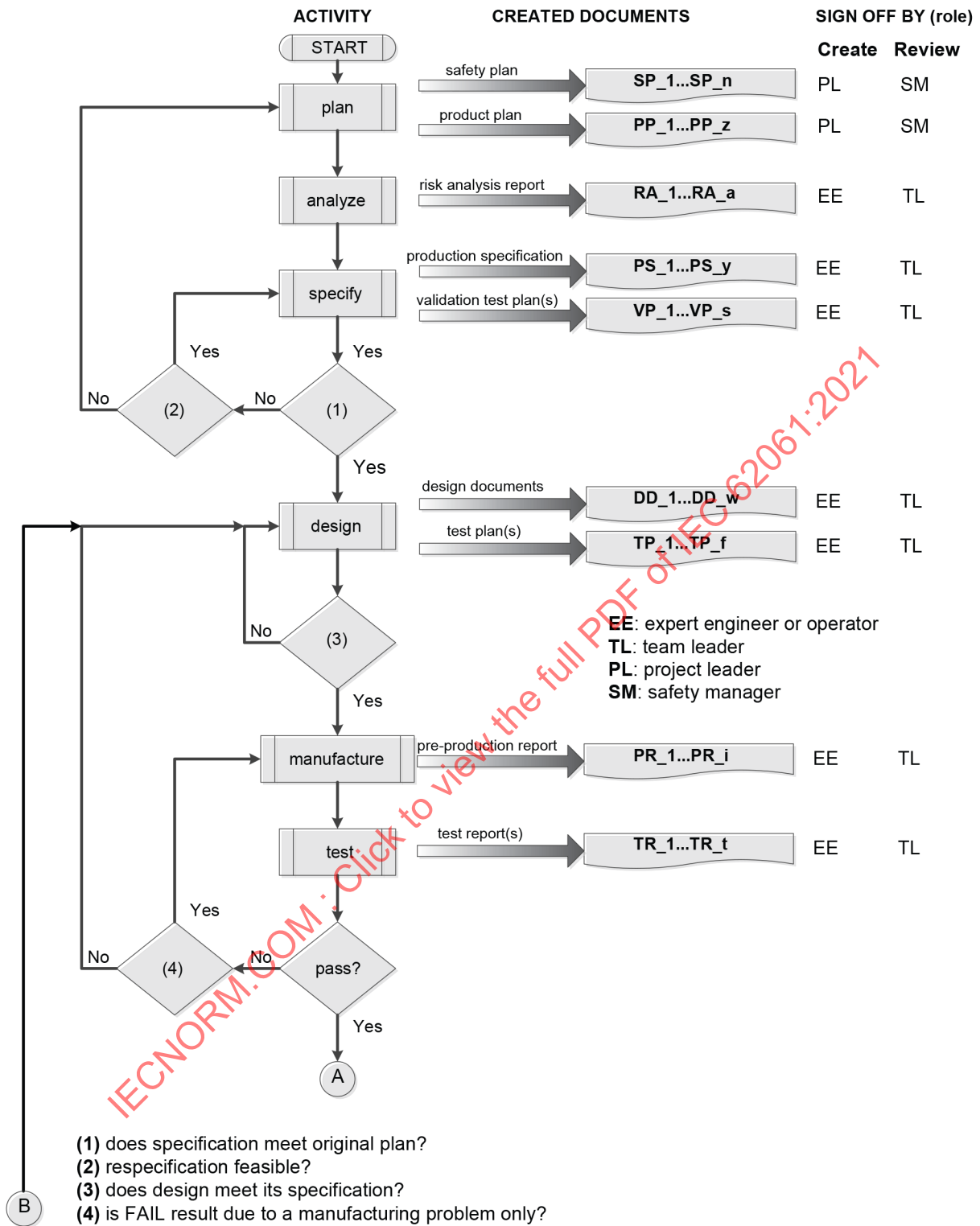


Figure I.2 – Example of activities, documents and roles (1 of 2)

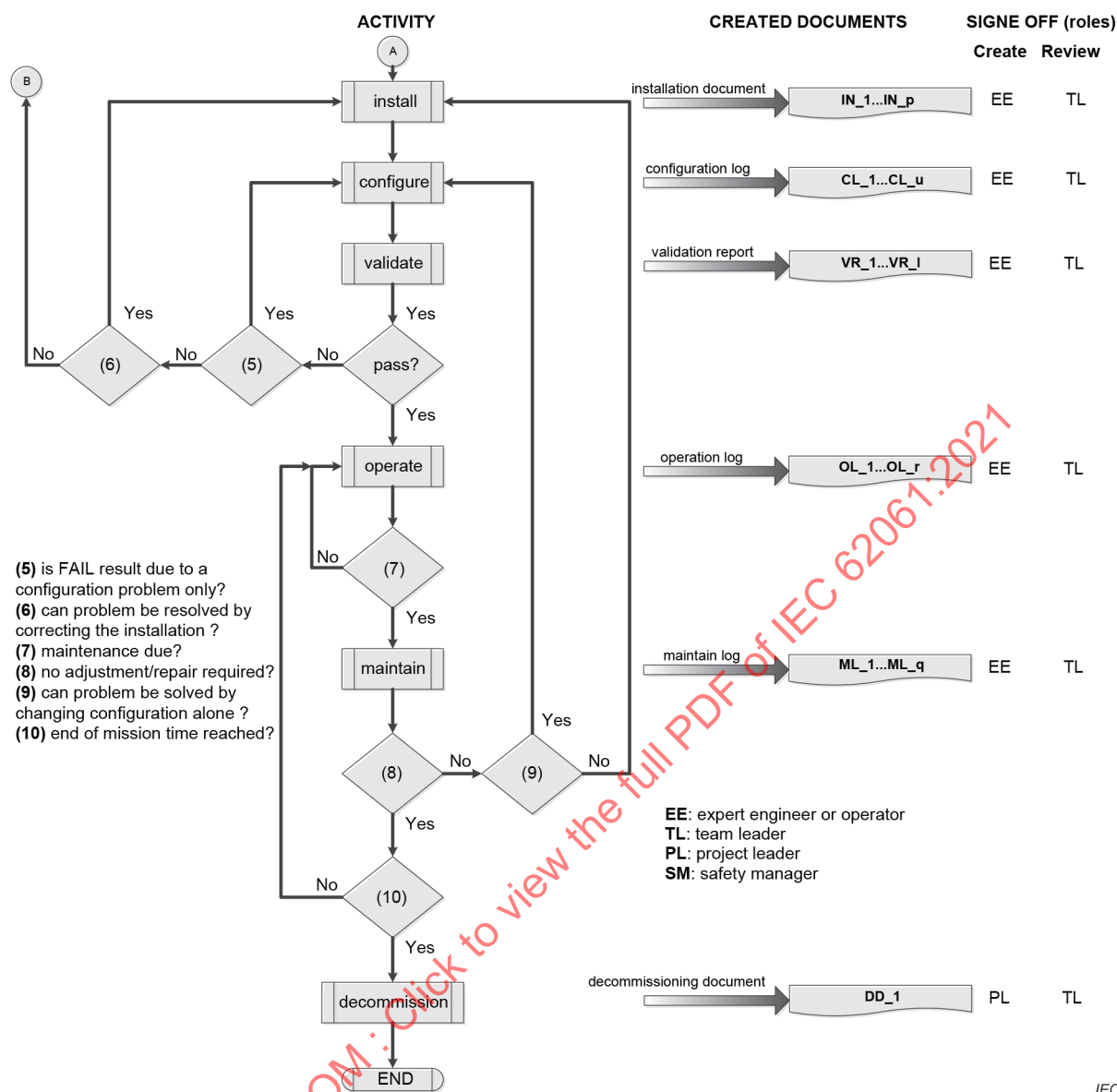


Figure I.2 (2 of 2)

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.

IECNORM.COM : Click to view the full PDF of IEC 62061:2021

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 60449:1973¹, *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, *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 63074:2019, *Safety of machinery – Security aspects related to 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, *Information technology – Security techniques – 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

IECNORM.COM : Click to view the full PDF of IEC 62061:2021

SOMMAIRE

AVANT-PROPOS	151
INTRODUCTION.....	153
1 Domaine d'application	154
2 Références normatives	155
3 Termes, définitions et abréviations	156
3.1 Liste alphabétique des définitions	156
3.2 Termes et définitions	157
3.3 Abréviations.....	172
4 Processus de conception d'un SCS et gestion de la sécurité fonctionnelle.....	173
4.1 Objectifs	173
4.2 Processus de conception	173
4.3 Gestion de la sécurité fonctionnelle à l'aide d'un plan de sécurité fonctionnelle	175
4.4 Gestion de configuration	177
4.5 Modification	177
5 Spécification d'une fonction de sécurité	178
5.1 Objectifs	178
5.2 Spécification des exigences de sécurité (SRS)	178
5.2.1 Généralités	178
5.2.2 Informations à mettre à disposition	178
5.2.3 Spécification des exigences fonctionnelles	179
5.2.4 Estimation du mode de fonctionnement à sollicitation	180
5.2.5 Spécification des exigences d'intégrité de sécurité	180
6 Conception d'un SCS.....	181
6.1 Généralités	181
6.2 Architecture de sous-système en fonction de la décomposition descendante	181
6.3 Méthodologie de base – Utilisation du sous-système	181
6.3.1 Généralités.....	181
6.3.2 Décomposition du SCS	182
6.3.3 Attribution de sous-fonction	184
6.3.4 Utilisation d'un sous-système type	184
6.4 Détermination de l'intégrité de sécurité du SCS	184
6.4.1 Généralités	184
6.4.2 PFH.....	185
6.5 Exigences pour l'intégrité de sécurité systématique du SCS	185
6.5.1 Exigences pour l'évitement des défaillances systématiques du matériel.....	185
6.5.2 Exigences pour la maîtrise des anomalies systématiques	186
6.6 Immunité électromagnétique	187
6.7 Paramétrisation manuelle liée au logiciel	188
6.7.1 Généralités	188
6.7.2 Influences sur les paramètres relatifs à la sécurité	188
6.7.3 Exigences relatives à la paramétrisation manuelle liée au logiciel.....	189
6.7.4 Vérification de l'outil de paramétrisation	190
6.7.5 Performances de la paramétrisation manuelle liée au logiciel	190
6.8 Aspects liés à la sécurité	191
6.9 Aspects des essais périodiques	191

7	Conception et développement d'un sous-système.....	192
7.1	Généralités	192
7.2	Conception de l'architecture d'un sous-système.....	193
7.3	Exigences pour le choix et la conception du sous-système et des éléments de sous-système.....	194
7.3.1	Généralités	194
7.3.2	Intégrité systématique	194
7.3.3	Prise en considération et exclusion des anomalies	197
7.3.4	Taux de défaillance de l'élément de sous-système	199
7.4	Contraintes architecturales d'un sous-système	203
7.4.1	Généralités	203
7.4.2	Estimation de la proportion de défaillances en sécurité (<i>SFF</i>)	204
7.4.3	Comportement (du SCS) lors de la détection d'une anomalie dans un sous-système	205
7.4.4	Réalisation des fonctions de diagnostic	206
7.5	Architectures de conception du sous-système.....	207
7.5.1	Généralités	207
7.5.2	Architectures de sous-système simple	208
7.5.3	Exigences de base	209
7.6	Fréquence moyenne de défaillance dangereuse par heure (<i>PFH</i>) des sous-systèmes.....	210
7.6.1	Généralités	210
7.6.2	Méthodes d'estimation de la <i>PFH</i> d'un sous-système	210
7.6.3	Approche simplifiée pour l'estimation de la contribution des défaillances de cause commune (CCF)	211
8	Logiciels.....	211
8.1	Généralités	211
8.2	Définition des niveaux logiciels	211
8.3	Niveau logiciel 1	213
8.3.1	Cycle de vie de sécurité du logiciel – Niveau logiciel 1	213
8.3.2	Conception du logiciel – Niveau logiciel 1	214
8.3.3	Conception du module – Niveau logiciel 1.....	216
8.3.4	Codage – Niveau logiciel 1	217
8.3.5	Essai du module – Niveau logiciel 1.....	217
8.3.6	Essai du logiciel – Niveau logiciel 1	218
8.3.7	Documentation – Niveau logiciel 1	218
8.3.8	Processus de gestion de la configuration et des modifications – Niveau logiciel 1.....	219
8.4	Niveau logiciel 2	219
8.4.1	Cycle de vie de sécurité du logiciel – Niveau logiciel 2	219
8.4.2	Conception du logiciel – Niveau logiciel 2	221
8.4.3	Conception du système logiciel – Niveau logiciel 2	223
8.4.4	Conception du module – Niveau logiciel 2.....	224
8.4.5	Codage – Niveau logiciel 2	224
8.4.6	Essai du module – Niveau logiciel 2.....	225
8.4.7	Essai d'intégration du logiciel – Niveau logiciel 2	225
8.4.8	Essai du logiciel – Niveau logiciel 2	225
8.4.9	Documentation – Niveau logiciel 2	227
8.4.10	Processus de gestion de la configuration et des modifications – Niveau logiciel 2.....	227

9	Validation	228
9.1	Principes de validation	228
9.1.1	Plan de validation	230
9.1.2	Utilisation des listes d'anomalies génériques	230
9.1.3	Listes d'anomalies spécifiques	230
9.1.4	Informations pour la validation	231
9.1.5	Consignation de la validation	231
9.2	Analyse dans le cadre de la validation	232
9.2.1	Généralités	232
9.2.2	Techniques d'analyse	232
9.2.3	Vérification de la spécification des exigences de sécurité (SRS)	232
9.3	Essais dans le cadre de la validation	233
9.3.1	Généralités	233
9.3.2	Exactitude de mesure	233
9.3.3	Exigences plus strictes	234
9.3.4	Échantillons d'essai	234
9.4	Validation de la fonction de sécurité	234
9.4.1	Généralités	234
9.4.2	Analyses et essais	235
9.5	Validation de l'intégrité de sécurité du SCS	235
9.5.1	Généralités	235
9.5.2	Validation du ou des sous-systèmes	236
9.5.3	Validation des mesures contre les défaillances systématiques	236
9.5.4	Validation du logiciel relatif à la sécurité	236
9.5.5	Validation de la combinaison de sous-systèmes	237
10	Documentation	238
10.1	Généralités	238
10.2	Documentation technique	238
10.3	Informations pour l'utilisation du SCS	239
10.3.1	Généralités	239
10.3.2	Informations relatives à l'utilisation données par le fabricant de sous-systèmes	240
10.3.3	Informations relatives à l'utilisation données par l'intégrateur du SCS	241
Annexe A (informative)	Détermination de l'intégrité de sécurité exigée	242
A.1	Généralités	242
A.2	Attribution de matrice pour le niveau de SIL exigé	242
A.2.1	Signalisation/Identification d'un phénomène dangereux	242
A.2.2	Estimation du risque	242
A.2.3	Sévérité (Se)	243
A.2.4	Probabilité d'apparition d'un dommage	244
A.2.5	Classe de probabilité d'un dommage (CI)	247
A.2.6	Attribution du niveau de SIL	247
A.3	Chevauchement de phénomènes dangereux	249
Annexe B (informative)	Exemple de méthodologie de conception de SCS	250
B.1	Généralités	250
B.2	Spécification des exigences de sécurité	250
B.3	Décomposition de la fonction de sécurité	251
B.4	Conception du SCS à l'aide de sous-systèmes	251
B.4.1	Généralités	251

B.4.2	Conception du sous-système 1 – "surveillance de la porte de protection"	252
B.4.3	Conception du sous-système 2 – "logique d'évaluation"	254
B.4.4	Conception du sous-système 3 – "commande de moteur"	254
B.4.5	Évaluation du SCS	255
B.4.6	PFH	255
B.5	Vérification	256
B.5.1	Généralités	256
B.5.2	Analyse	256
B.5.3	Essais	256
Annexe C (informative)	Exemples de valeurs $MTTF_D$ pour des composants simples	257
C.1	Généralités	257
C.2	Méthode reposant sur le respect des règles de l'art	257
C.3	Composants hydrauliques	257
C.4	$MTTF_D$ des composants pneumatiques, mécaniques et électromécaniques	258
Annexe D (informative)	Exemples de couverture du diagnostic (DC)	260
Annexe E (informative)	Méthodologie pour l'estimation de la sensibilité aux défaillances de cause commune (CCF)	263
E.1	Généralités	263
E.2	Méthodologie	263
E.2.1	Exigences pour la CCF	263
E.2.2	Estimation des effets de la CCF	263
Annexe F (informative)	Lignes directrices relatives au niveau logiciel 1	266
F.1	Exigences de sécurité du logiciel	266
F.2	Lignes directrices en matière de codage	267
F.3	Spécification des fonctions de sécurité	269
F.4	Spécification de la conception du matériel	271
F.5	Spécification de conception du système logiciel	272
F.6	Protocoles	276
Annexe G (informative)	Exemples de fonctions de sécurité	278
Annexe H (informative)	Approches simplifiées pour évaluer la valeur PFH d'un sous-système	279
H.1	Approche du tableau d'allocation	279
H.2	Formules simplifiées pour l'estimation de PFH	282
H.2.1	Généralités	282
H.2.2	Architecture A d'un sous-système simple: simple canal sans fonction de diagnostic	282
H.2.3	Architecture B d'un sous-système simple: double canal sans fonction de diagnostic	283
H.2.4	Architecture C d'un sous-système simple: simple canal avec fonction de diagnostic	283
H.2.5	Architecture D d'un sous-système simple: double canal avec fonction(s) de diagnostic	289
H.3	Méthode de comptage des parties	290
Annexe I (informative)	Plan de sécurité fonctionnelle et activités de conception	291
I.1	Généralités	291
I.2	Exemple de plan de conception d'une machine incluant un plan de sécurité	291
I.3	Exemple d'activités, de documents et de rôles	291

Annexe J (informative) Indépendance pour les activités d'examen et d'essai/de vérification/de validation	294
J.1 Conception de logiciels	294
J.2 Validation	294
Bibliographie	296
Figure 1 – Domaine d'application du présent document	155
Figure 2 – Intégration dans le processus de réduction du risque de l'ISO 12100 (extrait)	173
Figure 3 – Processus itératif de conception du système de commande relatif à la sécurité	174
Figure 4 – Exemple de combinaison de sous-systèmes en un SCS	175
Figure 5 – Définition possible par hypothèse d'un mode à forte sollicitation par l'activation d'une fonction de sécurité à faible sollicitation au moins une fois par an	180
Figure 6 – Exemples de décomposition classique d'une fonction de sécurité en sous- fonctions et de son attribution aux sous-systèmes	183
Figure 7 – Exemple d'intégrité de sécurité d'une fonction de sécurité reposant sur des sous-systèmes attribués en tant que SCS unique	185
Figure 8 – Représentation logique d'un sous-système de type A	208
Figure 9 – Représentation logique d'un sous-système de type B	208
Figure 10 – Représentation logique d'un sous-système de type C	208
Figure 11 – Représentation logique d'un sous-système de type D	209
Figure 12 – Modèle en V pour le niveau logiciel 1	213
Figure 13 – Modèle en V pour les modules logiciels personnalisés par le concepteur pour le niveau logiciel 1	214
Figure 14 – Modèle en V du cycle de vie de sécurité du logiciel pour le niveau logiciel 2	220
Figure 15 – Aperçu du processus de validation	229
Figure A.1 – Paramètres utilisés dans l'estimation du risque	242
Figure A.2 – Exemple de pro forma pour procédé d'attribution de SIL	249
Figure B.1 – Décomposition de la fonction de sécurité	251
Figure B.2 – Présentation de la conception des sous-systèmes du SCS	251
Figure F.1 – Croquis de l'usine	269
Figure F.2 – Conception de l'architecture modulaire principale	273
Figure F.3 – Approche de conception principale de l'évaluation logique	274
Figure F.4 – Exemple de représentation logique (croquis du programme)	275
Figure H.1 – Représentation logique d'un sous-système de type A	282
Figure H.2 – Représentation logique d'un sous-système de type B	283
Figure H.3 – Représentation logique d'un sous-système de type C	283
Figure H.4 – Corrélation entre le sous-système de type C et la fonction de traitement des anomalies correspondante	284
Figure H.5 – Sous-système de type C avec fonction externe de traitement des anomalies	285
Figure H.6 – Sous-système de type C avec diagnostics externes des anomalies	286
Figure H.7 – Sous-système de type C avec réaction externe à l'anomalie	286
Figure H.8 – Sous-système de type C avec diagnostics internes des anomalies et réaction interne à l'anomalie	287

Figure H.9 – Représentation logique d'un sous-système de type D	289
Figure I.1 – Exemple de plan de conception d'une machine incluant un plan de sécurité.....	291
Figure I.2 – Exemple d'activités, de documents et de rôles (1 sur 2)	292
Tableau 1 – Termes utilisés dans l'IEC 62061.....	156
Tableau 2 – Abréviations utilisées dans l'IEC 62061	172
Tableau 3 – SIL et limites des valeurs de PFH	181
Tableau 4 – SIL exigé et PFH du sous-système type.....	184
Tableau 5 – Informations pertinentes pour chaque sous-système	193
Tableau 6 – Contraintes architecturales sur un sous-système: SIL maximal pouvant être revendiqué pour un SCS utilisant ce sous-système.....	204
Tableau 7 – Aperçu des exigences de base et de l'interrelation avec les architectures de sous-système simple.....	210
Tableau 8 – Différents niveaux de logiciels d'application.....	212
Tableau 9 – Documentation d'un SCS.....	239
Tableau A.1 – Classification de la sévérité (Se)	243
Tableau A.2 – Classification de la fréquence et durée de l'exposition (Fr).....	244
Tableau A.3 – Classification de la probabilité (Pr).....	246
Tableau A.4 – Classification de la probabilité d'évitement ou de limitation d'un dommage (Av)	246
Tableau A.5 – Paramètres utilisés pour déterminer la classe de probabilité d'un dommage (CI).....	247
Tableau A.6 – Attribution de matrice pour déterminer le niveau de SIL exigé (ou PL_r) pour une fonction de sécurité.....	248
Tableau B.1 – Spécification des exigences de sécurité – exemple de présentation	250
Tableau B.2 – Intégrité systématique – exemple de présentation	255
Tableau B.3 – Vérification par des essais	256
Tableau C.1 – Normes de référence et valeurs $MTTF_D$ ou B_{10D} des composants.....	258
Tableau D.1 – Estimations de la couverture du diagnostic (DC)(1 sur 2)	261
Tableau E.1 – Critères d'estimation des CCF.....	264
Tableau E.2 – Critères d'estimation des CCF.....	265
Tableau F.1 – Exemple de documents pertinents relatifs au modèle en V simplifié	266
Tableau F.2 – Exemples de lignes directrices en matière de codage.....	268
Tableau F.3 – Fonctions de sécurité spécifiées.....	270
Tableau F.4 – Liste des signaux d'entrée et de sortie	272
Tableau F.5 – Exemple de matrice de cause et effet simplifiée	275
Tableau F.6 – Vérification de la spécification de conception du système logiciel.....	276
Tableau F.7 – Revue de code du logiciel	276
Tableau F.8 – Validation du logiciel	277
Tableau G.1 – Exemples de fonctions de sécurité classiques.....	278
Tableau H.1 – Allocation de la valeur PFH d'un sous-système	280
Tableau H.2 – Relations entre B_{10D} , les opérations et $MTTF_D$	281
Tableau H.3 – Valeur minimale de $1/\lambda_{D FH}$ pour l'applicabilité de l'équation PFH (H.4)	287

Tableau J.1 – Niveaux minimaux d'indépendance pour les activités d'examen, d'essai et de vérification	294
Tableau J.2 – Niveaux minimaux d'indépendance pour les activités de validation	295

IECNORM.COM : Click to view the full PDF of IEC 62061:2021

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

**SÉCURITÉ DES MACHINES –
SÉCURITÉ FONCTIONNELLE DES SYSTÈMES
DE COMMANDE RELATIFS À LA SÉCURITÉ****AVANT-PROPOS**

- 1) La Commission Électrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. À cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications. L'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

L'IEC 62061 a été établie par le comité d'études 44 de l'IEC: Sécurité des machines – Aspects électrotechniques. Il s'agit d'une Norme internationale.

Cette deuxième édition annule et remplace la première édition parue en 2005, l'Amendement 1:2012 ainsi que l'Amendement 2:2015. Cette édition constitue une révision technique.

Cette édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente:

- la structure a été modifiée et le contenu a été mis à jour pour refléter le processus de conception de la fonction de sécurité,
- la norme a été étendue aux technologies non électriques,
- définitions mises à jour pour être alignées sur l'IEC 61508-4,

- plan de sécurité fonctionnelle introduit et gestion de configuration mise à jour (Article 4),
- exigences relatives au paramétrage étendues (Article 6),
- référence aux exigences relatives à la sécurité ajoutée (Paragraphe 6.8)
- exigences relatives aux essais périodiques ajoutées (Paragraphe 6.9),
- différentes améliorations et clarifications relatives aux architectures et aux calculs de fiabilité (Article 6 et Article 7),
- décalage entre le "SILCL" et le "SIL maximal" d'un sous-système (Article 7),
- cas d'utilisation pour les logiciels décrits, y compris les exigences (Article 8),
- exigences relatives à l'indépendance des activités de vérification (Article 8) et de validation (Article 9) du logiciel ajoutées,
- nouvelle annexe informative avec des exemples (Annex G),
- nouvelles annexes informatives relatives aux valeurs $MTTF_D$, aux diagnostics et aux méthodes de calcul des architectures (Annex C, Annex D et Annex H).

Le texte de cette Norme internationale est issu des documents suivants:

Projet	Rapport de vote
44/885/FDIS	44/888/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à son approbation.

La langue employée pour l'élaboration de cette Norme internationale est l'anglais.

Ce document a été rédigé selon les Directives ISO/IEC, Partie 2, il a été développé selon les Directives ISO/IEC, Partie 1 et les Directives ISO/IEC, Supplément IEC, disponibles sous www.iec.ch/members_experts/refdocs. Les principaux types de documents développés par l'IEC sont décrits plus en détail sous www.iec.ch/standardsdev/publications.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. À cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer ce document en utilisant une imprimante couleur.

INTRODUCTION

Par suite de l'automatisation, ainsi que de la demande d'une production plus élevée avec une réduction des efforts physiques des opérateurs, les systèmes de commande relatifs à la sécurité (appelés SCS ci-après) des machines jouent un rôle croissant dans la réalisation de la sécurité d'ensemble des machines. De ce fait, les SCS eux-mêmes utilisent de plus en plus souvent une technologie électronique complexe.

L'IEC 62061 spécifie les exigences pour la conception et la réalisation des systèmes de commande des machines relatifs à la sécurité. Le présent document est spécifique au secteur des machines dans le cadre de l'IEC 61508.

NOTE Bien que l'IEC 62061 et l'ISO 13849-1 utilisent des méthodologies différentes en matière de conception des systèmes de commande relatifs à la sécurité, elles visent à réaliser le même objectif de réduction de risque.

La présente Norme internationale est destinée à être utilisée par les concepteurs de machines, les fabricants et les intégrateurs de systèmes de commande, et autres impliqués dans la spécification, la conception et la validation d'un SCS. Elle présente une approche et donne les exigences nécessaires à la réalisation du fonctionnement requis et facilite la spécification des fonctions de sécurité destinées à réaliser la réduction de risque.

Le présent document donne un cadre spécifique au secteur des machines pour la sécurité fonctionnelle d'un SCS de machine. Il couvre uniquement les aspects du cycle de vie de sécurité relatifs à l'allocation des exigences de sécurité jusqu'à la validation de la sécurité. Des exigences sont données pour information pour une utilisation sûre des SCS de machines, lesquelles peuvent aussi être appropriées pour des phases ultérieures de la vie d'un SCS.

Il existe de nombreuses circonstances dans les machines où les SCS sont utilisés comme partie des mesures de sécurité développées pour réaliser la réduction de risque. Un exemple typique est l'utilisation d'un protecteur avec dispositif de verrouillage qui, lorsqu'il est ouvert pour permettre l'accès à la zone dangereuse, signale aux parties relatives à la sécurité du système de commande de la machine d'arrêter le fonctionnement dangereux de la machine. En automatisation, le système de commande de la machine utilisé pour réaliser le fonctionnement correct du processus machine contribue souvent à la sécurité en réduisant les risques associés aux phénomènes dangereux résultant directement de défaillances du système de commande. Le présent document donne une méthodologie et les exigences pour:

- assigner le niveau d'intégrité de sécurité exigé pour chaque fonction de sécurité devant être mise en œuvre par les SCS;
- permettre la conception des SCS appropriés à la ou aux fonctions de commande assignées relatives à la sécurité;
- intégrer les sous-systèmes relatifs à la sécurité conçus selon d'autres normes applicables relatives à la sécurité fonctionnelle (voir 6.3.4);
- valider les SCS.

Le présent document est destiné à être utilisé dans le cadre de la réduction systématique du risque, conjointement avec l'appréciation du risque décrite dans l'ISO 12100. Les méthodologies conseillées pour l'attribution d'intégrité de sécurité sont données dans l'Annex A informative.

SÉCURITÉ DES MACHINES – SÉCURITÉ FONCTIONNELLE DES SYSTÈMES DE COMMANDE RELATIFS À LA SÉCURITÉ

1 Domaine d'application

La présente Norme internationale spécifie les exigences et donne des recommandations pour la conception, l'intégration et la validation des systèmes de commande relatifs à la sécurité (SCS) pour les machines. Elle s'applique aux systèmes de commande utilisés, séparément ou en combinaison, pour assurer les fonctions de sécurité de machines qui ne sont pas portables à la main en fonctionnement, y compris un groupe de machines fonctionnant ensemble d'une manière coordonnée.

Le présent document est spécifique au secteur des machines dans le cadre de l'IEC 61508 (toutes les parties).

La conception de sous-systèmes ou d'éléments de sous-système électroniques programmables complexes ne relève pas du domaine d'application du présent document. Ces éléments relèvent du domaine d'application de l'IEC 61508 ou de normes qui lui sont associées (voir la Figure 1).

NOTE 1 Les éléments tels que les systèmes sur puce ou les cartes de microcontrôleur sont considérés comme des sous-systèmes électroniques programmables complexes.

Le corps principal de la présente norme sectorielle spécifie les exigences générales en matière de conception et de vérification d'un système de commande relatif à la sécurité destiné à être utilisé en mode à forte sollicitation/continu.

Le présent document:

- ne concerne que les exigences de sécurité fonctionnelle destinées à réduire le risque de situations dangereuses;
- se limite aux risques résultant directement des phénomènes dangereux de la machine elle-même ou d'un groupe de machines fonctionnant ensemble d'une manière coordonnée;

NOTE 2 Les exigences pour réduire les risques provenant d'autres phénomènes dangereux sont données dans les normes sectorielles appropriées. Par exemple, pour une ou plusieurs machines qui font partie d'une activité processus, des informations supplémentaires sont disponibles dans l'IEC 61511.

Le présent document ne concerne pas

- les phénomènes dangereux électriques provenant du matériel de commande électrique lui-même (par exemple choc électrique – voir l'IEC 60204-1);
- les autres exigences relatives à la sécurité nécessaires au niveau de la machine (la protection par protecteur, par exemple);
- les mesures particulières pour les aspects liés à la sécurité – voir l'IEC TR 63074.

Le présent document n'est pas destiné à limiter ou inhiber les progrès technologiques.

La Figure 1 donne une représentation du domaine d'application du présent document.

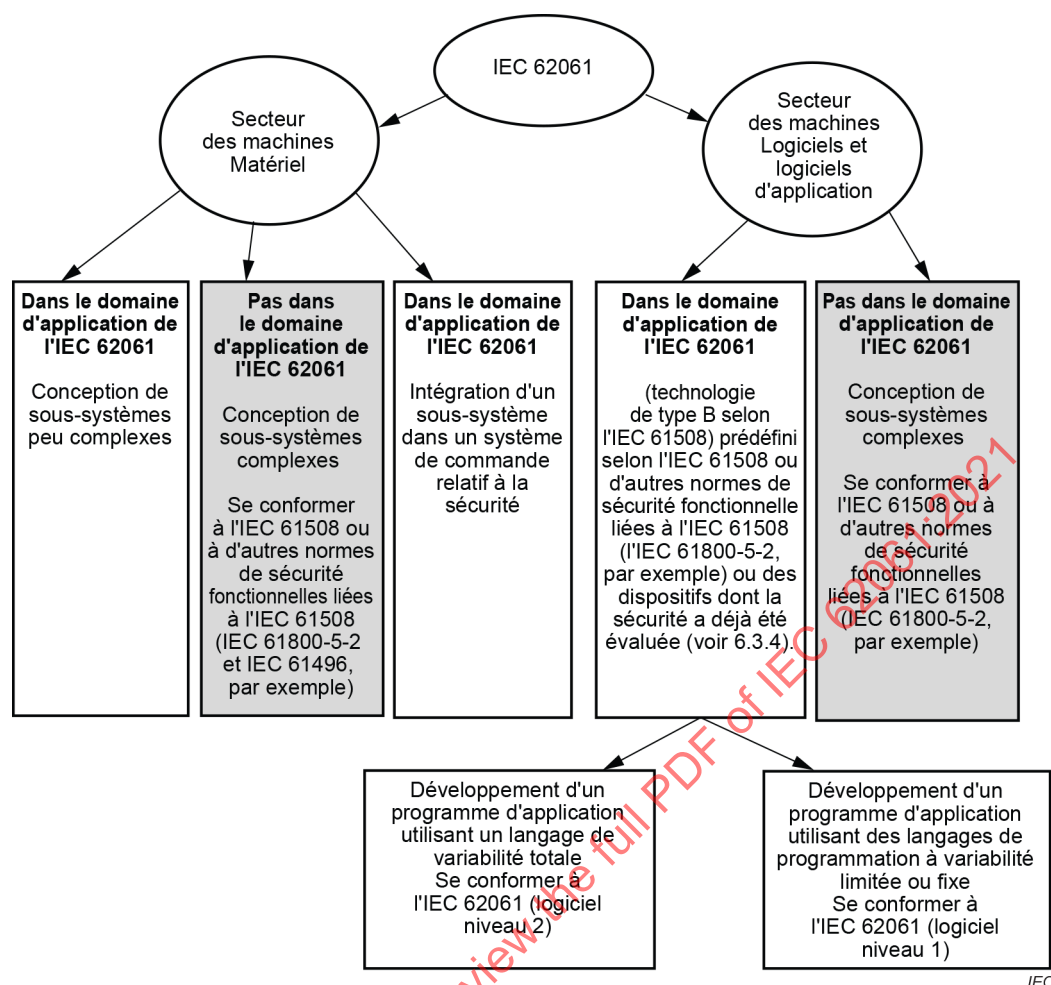


Figure 1 – Domaine d'application du présent document

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 60204-1:2016, *Sécurité des machines – Équipement électrique des machines – Partie 1: Exigences générales*

IEC 61000-1-2:2016, *Compatibilité électromagnétique (CEM) – Partie 1-2: Généralités – Méthodologie pour la réalisation de la sécurité fonctionnelle des systèmes électriques et électroniques, y compris les équipements, du point de vue des phénomènes électromagnétiques*

IEC 61508 (toutes les parties), *Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité*

IEC 61508-2:2010, *Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité – Partie 2: Exigences pour les systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité*

IEC 61508-3:2010, *Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité – Partie 3: Exigences concernant les logiciels*

ISO 12100:2010, *Sécurité des machines – Principes généraux de conception – Appréciation du risque et réduction du risque*

ISO 13849 (toutes les parties), *Sécurité des machines – Parties des systèmes de commande relatives à la sécurité*

ISO 13849-1:2015, *Sécurité des machines – Parties des systèmes de commande relatives à la sécurité – Partie 1: Principes généraux de conception*

ISO 13849-2:2012, *Sécurité des machines – Parties des systèmes de commande relatives à la sécurité – Partie 2: Validation*

3 Termes, définitions et abréviations

3.1 Liste alphabétique des définitions

Le Tableau 1 présente les termes utilisés dans l'IEC 62061, ainsi que quelques abréviations courantes relatives à la sécurité des machines.

Tableau 1 – Termes utilisés dans l'IEC 62061

Terme	Numéro de définition
logiciel d'application	3.2.59
contrainte architecturale	3.2.46
architecture	3.2.45
fréquence moyenne de défaillance dangereuse par heure (PFH)	3.2.29
probabilité moyenne de défaillance dangereuse en cas de sollicitation (PFD_{avg})	3.2.31
référentiel (de configuration)	3.2.67
dérivation	3.2.17
défaillance de cause commune (CCF)	3.2.56
composant complexe	3.2.8
gestion de configuration	3.2.66
mode continu	3.2.28
défaillance dangereuse	3.2.52
sollicitation	3.2.25
couverture du diagnostic (DC)	3.2.49
intervalle entre essais de diagnostic	3.2.50
logiciel intégré logiciel embarqué	3.2.60
défaillance	3.2.51
anomalie	3.2.33
tolérance aux anomalies	3.2.34
langage de variabilité totale (FVL)	3.2.61
sécurité fonctionnelle	3.2.10
tolérance aux anomalies du matériel (HFT)	3.2.35
intégrité de sécurité du matériel	3.2.22
dommage	3.2.12
phénomène dangereux	3.2.11
mode sollicitation élevée	3.2.27
intégrateur	3.2.13

Terme	Numéro de définition
langage de variabilité limitée (LVL)	3.2.62
composant de faible complexité	3.2.7
mode faible sollicitation	3.2.26
système de commande de la machine	3.2.2
machine	3.2.1
durée moyenne de réparation (MRT)	3.2.40
durée moyenne de fonctionnement avant défaillance ($MTTF$)	3.2.37
durée moyenne de fonctionnement avant défaillance dangereuse ($MTTF_D$)	3.2.38
durée moyenne de rétablissement (MTTR)	3.2.39
inhibition	3.2.16
type (SCS ou sous-système)	3.2.5
probabilité de défaillance dangereuse en cas de sollicitation (PFD)	3.2.30
temps de sécurité du processus	3.2.41
couverture périodique d'essai	3.2.48
essai périodique	3.2.47
mesure de prévention	3.2.14
défaillance aléatoire du matériel	3.2.57
rapport de défaillance dangereuse (RDF)	3.2.55
risque	3.2.15
défaillance en sécurité	3.2.53
proportion de défaillances en sécurité (SFF)	3.2.54
état de sécurité	3.2.68
sécurité	3.2.9
fonction de sécurité	3.2.18
intégrité de sécurité	3.2.21
niveau d'intégrité de sécurité (SIL)	3.2.24
système de commande relatif à la sécurité (SCS)	3.2.3
logiciel relatif à la sécurité	3.2.63
sécurité	3.2.69
fonction diagnostic (du SCS)	3.2.19
fonction réaction à l'anomalie (du SCS)	3.2.20
sous-système	3.2.4
élément de sous-système	3.2.6
sous-fonction	3.2.36
défaillance systématique	3.2.58
intégrité de sécurité systématique	3.2.23
objectif chiffre de défaillance	3.2.32
durée de fonctionnement utile	3.2.42
validation (de la fonction de sécurité)	3.2.65
vérification	3.2.64
composant éprouvé	3.2.43
principes de sécurité éprouvés	3.2.44

3.2 Termes et définitions

Pour les besoins du présent document, les termes et définitions suivants s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse <http://www.electropedia.org/>

- ISO Online browsing platform: disponible à l'adresse <http://www.iso.org/obp>

3.2.1

machine

ensemble équipé ou destiné à être équipé d'un système d'entraînement, composé de pièces ou d'organes liés entre eux dont au moins un est mobile et qui sont réunis de façon solidaire en vue d'une application définie

Note 1 à l'article: Le terme "machine" désigne aussi un ensemble de machines qui, afin de concourir à un même résultat, sont disposées et commandées de manière à être solidaires dans leur fonctionnement.

[SOURCE: ISO 12100:2010, 3.1]

3.2.2

système de commande de la machine

système qui réagit à des signaux d'entrée provenant de la machine et/ou d'un opérateur et qui produit des signaux de sortie qui font que la machine fonctionne de la façon souhaitée

Note 1 à l'article: Le système de commande de la machine comprend des dispositifs d'entrée et des éléments terminaux.

[SOURCE: IEC 61508-4:2010, 3.3.3, modifiée – le terme défini a été modifié, "processus" a été remplacé par "machine"]

3.2.3

système de commande relatif à la sécurité

SCS

partie du système de commande d'une machine qui met en œuvre une fonction de sécurité par un ou plusieurs sous-systèmes

Note 1 à l'article: Le SCS s'apparente au SRECS la précédente édition du présent document.

Note 2 à l'article: L'abréviation "SCS" est dérivée du terme anglais développé correspondant "*safety-related control system*".

3.2.4

sous-système

entité de la conception architecturale générale d'un système relatif à la sécurité dans laquelle une défaillance dangereuse du sous-système conduit à une défaillance dangereuse d'une fonction de sécurité

Note 1 à l'article: Cette définition diffère du langage courant où le terme "sous-système" peut signifier une quelconque partie subdivisée d'une entité; le terme "sous-système" est utilisé dans le présent document dans une hiérarchie terminologique bien déterminée: le "sous-système" est le premier niveau de subdivision d'un système. Les parties résultant de subdivisions ultérieures d'un sous-système sont appelées "éléments de sous-système".

Note 2 à l'article: Un sous-système complet peut être constitué d'un certain nombre d'éléments de sous-système identifiables et séparés.

Note 3 à l'article: La spécification du sous-système inclut son rôle dans la fonction de sécurité et son interface avec les autres sous-systèmes du SCS.

Note 4 à l'article: Un sous-système peut faire partie de plusieurs fonctions de sécurité, par exemple, la même combinaison de contacteurs peut être utilisée pour couper l'alimentation d'un moteur soit en cas de détection d'une personne dans une zone de danger, soit également en cas d'ouverture d'un protecteur avec dispositif de verrouillage.

[SOURCE: IEC 61508-4:2010, 3.4.4, modifiée – les renvois sont supprimés et les notes ajoutées]

3.2.5

SCS ou sous-système type

SCS ou sous-système satisfaisant aux exigences correspondantes d'une norme de sécurité fonctionnelle

3.2.6

élément de sous-système

partie d'un sous-système comprenant un composant unique ou un quelconque groupe de composants

Note 1 à l'article: Un élément de sous-système peut être composé du matériel et des logiciels.

Note 2 à l'article: Les éléments qui ne sont pas directement nécessaires pour la fonction de sécurité, mais pouvant la prendre en charge, ne sont pas inclus (éléments de filtres, protection contre les surtensions, par exemple).

Note 3 à l'article: Un élément de sous-système est le niveau le plus bas de détail à prendre en considération pour assurer que les exigences d'une sous-fonction sont satisfaites.

3.2.7

composant/sous-système de faible complexité

composant/sous-système pour lequel

- les modes de défaillance sont bien définis; et
- le comportement en conditions d'anomalie peut être complètement déterminé

Note 1 à l'article: Le comportement du composant/sous-système de faible complexité dans des conditions d'anomalie peut être déterminé par des méthodes analytiques et/ou d'essai.

Note 2 à l'article: Les interrupteurs de fin de course, les relais électromécaniques ou les contacteurs sont des exemples de composant/sous-système de faible complexité.

[SOURCE: IEC 61508-4:2010, 3.4.3, modifiée – le terme défini a été modifié, ce qui conduit à une reformulation du texte. L'Exemple est converti en note 2]

3.2.8

composant/sous-système complexe

composant/sous-système dans lequel

- les modes de défaillance ne sont pas bien définis; ou
- le comportement en conditions d'anomalie ne peut être complètement déterminé

3.2.9

sécurité

absence de risque inacceptable

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

3.2.10

sécurité fonctionnelle

sous-ensemble de la sécurité globale de la machine et du système de commande de la machine qui dépend du fonctionnement correct du SCS et d'autres mesures de réduction du risque

[SOURCE: IEC 61508-4:2010, 3.1.12, modifiée – utilisation des termes machine, système de commande de la machine et SCS]

3.2.11

phénomène dangereux

source potentielle de dommage

Note 1 à l'article: Le terme "phénomène dangereux" peut être qualifié de manière à faire apparaître son origine ou la nature du dommage potentiel (par exemple phénomène dangereux de choc électrique, phénomène dangereux d'écrasement, phénomène dangereux de coupure, phénomène dangereux d'intoxication, phénomène dangereux d'incendie).

[SOURCE: ISO 12100:2010, 3.6, modifiée – la Note 1 a été modifiée et les Notes 2 et 3 supprimées]

3.2.12

dommage

blessure physique ou atteinte à la santé des personnes

[SOURCE: ISO/IEC Guide 51:2014, 3.1, modifiée – "ou atteinte aux biens ou à l'environnement" supprimé]

3.2.13

intégrateur

entité qui conçoit, fabrique ou assemble un système de fabrication intégré et qui est responsable de la stratégie de sécurité, y compris les mesures de prévention, les interfaces de commande et les interconnexions du système de commande

Note 1 à l'article: L'intégrateur peut être par exemple un fabricant, un assembleur, une société d'ingénierie, ou une entité ayant l'entière responsabilité de la machine.

[SOURCE: ISO 11161:2007, 3.10, modifiée – "fournit" a été supprimé, la note a été reformulée]

3.2.14

mesure de prévention

mesure destinée à réduire le risque

[SOURCE: ISO 12100:2010, 3.19, modifiée – la liste à puces supprimée]

3.2.15

risque

combinaison de la probabilité de la survenue d'un dommage et de sa gravité

[SOURCE: ISO/IEC Guide 51:2014, 3.9 modifiée – note à l'article supprimée]

3.2.16

inhibition

interruption automatique temporaire d'une ou de plusieurs fonctions de sécurité

Note 1 à l'article: D'autres moyens sont utilisés pour maintenir le niveau de sécurité.

[SOURCE: ISO 13849-1:2015, 3.1.8, modifiée – "par les SRP/CS" a été supprimé, note ajoutée]

3.2.17

dérivation

action ou installation empêchant l'exécution de tout ou partie des fonctionnalités du SCS

Note 1 à l'article: Des exemples d'inhibition incluent ce qui suit:

- le signal d'entrée en provenance du circuit de déclenchement est bloqué, mais les paramètres d'entrée et l'alarme sont toujours présentés à l'opérateur;
- le signal de sortie du circuit de déclenchement à un élément terminal est maintenu à l'état normal, empêchant le fonctionnement de l'élément terminal;
- une ligne de dérivation physique est fournie autour de l'élément terminal;
- un état d'entrée présélectionné (par exemple: acheminer/arrêter le signal d'entrée) ou un ensemble de paramètres est forcé par l'intermédiaire d'un outil d'ingénierie (par exemple: dans le programme d'application).

Note 2 à l'article: La notion de dérivation est également rendue par d'autres termes (par exemple: neutralisation, mise en échec, désactivation, forçage, inhibition, shuntage ou blocage).

[SOURCE: IEC 61511-1:2016, 3.2.4, modifiée – SIS remplacé par SCS]

3.2.18

fonction de sécurité

fonction mise en œuvre par un SCS avec un niveau d'intégrité spécifié, prévue pour maintenir la condition de sécurité de la machine ou empêcher un accroissement immédiat du ou des risques en ce qui concerne un événement dangereux particulier

Note 1 à l'article: Ce terme est utilisé à la place de "fonction de commande relative à la sécurité (SRCF)" de l'IEC 62061:2015. Cette définition est différente de celle de l'ISO 12100, car le présent document porte sur la réduction du risque réalisée par le SCS.

Note 2 à l'article: En règle générale, une fonction de sécurité commence par la détection et l'évaluation d'un "événement déclencheur" et se termine par une sortie provoquant la réaction d'un "actionneur".

Note 3 à l'article: Les parties de la machine qui exécutent la ou les fonctions, par exemple la réaction d'un actionneur, peuvent également faire partie de la ou des fonctions de sécurité.

[SOURCE: IEC 61508-4:2010, 3.5.1, modifiée – terminologie adaptée aux machines, "dispositif externe de réduction de risque" supprimé, l'exemple supprimé, notes ajoutées]

3.2.19

fonction diagnostic (du SCS)

fonction prévue pour détecter les anomalies dans un SCS et déclencher une fonction réaction à l'anomalie spécifiée en cas de détection d'une anomalie

Note 1 à l'article: Cette fonction est prévue pour détecter des anomalies qui sont susceptibles d'entraîner une défaillance dangereuse d'une fonction de sécurité et initier une fonction réaction à l'anomalie déterminée.

3.2.20

fonction réaction à l'anomalie (du SCS)

fonction qui est initiée lorsqu'une anomalie à l'intérieur d'un SCS est détectée par la fonction diagnostic du SCS

3.2.21

intégrité de sécurité

probabilité pour qu'un SCS ou son sous-système exécute de manière satisfaisante les fonctions de sécurité exigées dans toutes les conditions spécifiées et dans une période spécifiée

Note 1 à l'article: Plus le niveau d'intégrité de sécurité de l'entité est élevé, plus la probabilité d'une défaillance de cette entité dans l'exécution de la fonction relative à la sécurité exigée est faible.

Note 2 à l'article: L'intégrité de sécurité comprend l'intégrité de sécurité du matériel ainsi que l'intégrité de sécurité systématique.

[SOURCE: IEC 61508-4:2010, 3.5.4, modifiée – terminologie adaptée aux machines, notes 2, 3, 5 supprimées]

3.2.22

intégrité de sécurité du matériel

partie de l'intégrité de sécurité d'un SCS ou de ses sous-systèmes qui se rapporte aux défaillances aléatoires du matériel dans un mode dangereux de défaillance

Note 1 à l'article: Le terme se rapporte aux défaillances dans un mode dangereux, c'est-à-dire les défaillances d'un système relatif à la sécurité qui détériorent son intégrité de sécurité.

Note 2 à l'article: L'intégrité de sécurité du matériel inclut les contraintes architecturales.

[SOURCE: IEC 61508-4:2010, 3.5.7 – terminologie adaptée aux machines, note 1 abrégée, note 2 supprimée]

3.2.23

intégrité de sécurité systématique

partie de l'intégrité de sécurité d'un SCS ou de ses sous-systèmes qui se rapporte à sa résistance aux défaillances systématiques dans un mode dangereux

Note 1 à l'article: L'intégrité de sécurité systématique ne peut habituellement pas être quantifiée de manière précise.

Note 2 à l'article: Les exigences pour l'intégrité de sécurité systématique s'appliquent à la fois pour les aspects matériel et logiciel d'un SCS ou de ses sous-systèmes.

[SOURCE: IEC 61508-4:2010, 3.5.6, modifiée – terminologie adaptée aux machines, note 1 abrégée, note 2 ajoutée]

3.2.24

niveau d'intégrité de sécurité

SIL

niveau discret (parmi trois possibles) permettant de décrire la capacité à exécuter une fonction de sécurité lorsque le niveau trois d'intégrité de sécurité possède le plus haut degré d'intégrité et que le niveau un possède le plus bas

Note 1 à l'article: L'abréviation "SIL" est dérivée du terme anglais développé correspondant "*safety integrity level*".

3.2.25

sollicitation

événement qui conduit le SCS à exécuter une fonction de sécurité

Note 1 à l'article: Le mode sollicitation signifie qu'une fonction de sécurité ne peut être réalisée qu'après une requête (sollicitation) afin que la machine passe dans un état spécifié. Le SCS n'a aucune influence sur la machine tant qu'il n'y a pas de sollicitation de la fonction de sécurité.

Note 2 à l'article: Le taux de sollicitation (DR) ou la fréquence de sollicitations est l'un des principaux facteurs pris en considération lors de l'évaluation du mode de sollicitation, faible ou élevée. À cet effet particulier, le taux de sollicitation (DR) peut être identifié avec le taux d'événements, où le dommage se produit sans l'intervention de la fonction de sécurité. Ce taux peut être inférieur au taux réel de déclenchement de la fonction de sécurité pendant le fonctionnement.

Note 3 à l'article: Pour une fonction d'arrêt d'urgence, le mode sollicitation n'est pas défini. Pour déterminer le SIL atteint, le principe d'évaluation du mode sollicitation sélectionné des autres fonctions est en général applicable.

3.2.26

mode faible sollicitation

mode de fonctionnement dans lequel la fréquence des sollicitations d'une fonction de sécurité n'est pas supérieure à une par an

[SOURCE: IEC 61508-4:2010, 3.5.16, modifiée – issu de la définition de "mode de fonctionnement"]

3.2.27

mode sollicitation élevée

mode de fonctionnement dans lequel la fréquence des sollicitations d'une fonction de sécurité n'est pas supérieure à une par an

Note 1 à l'article: Le mode continu signifie qu'une fonction de sécurité est réalisée continuellement, c'est-à-dire que le SCS commande de façon continue la machine et qu'une défaillance (dangereuse) de sa fonction peut entraîner un phénomène dangereux.

Note 2 à l'article: La distinction entre le mode à forte sollicitation et le mode continu est pertinente pour la qualification des mesures de diagnostic (voir 7.4.3 et 7.4.4). Elle ne l'est pas pour l'objectif chiffré de défaillance et l'attribution d'un niveau de SIL.

[SOURCE: IEC 61508-4:2010, 3.5.16, modifiée – issu de la définition de "mode de fonctionnement", notes ajoutées]

3.2.28

mode continu

mode de fonctionnement dans lequel la fonction de sécurité maintient la machine dans un état sûr en fonctionnement normal

Note 1 à l'article: Le mode continu signifie qu'une fonction de sécurité est réalisée continuellement, c'est-à-dire que le SCS commande de façon continue la machine et qu'une défaillance (dangereuse) de sa fonction peut entraîner un phénomène dangereux.

Note 2 à l'article: La distinction entre le mode à forte sollicitation et le mode continu est pertinente pour la qualification des mesures de diagnostic (voir 7.4.3 et 7.4.4). Elle ne l'est pas pour l'objectif chiffré de défaillance et l'attribution d'un niveau de SIL.

[SOURCE: IEC 61508-4:2010, 3.5.16, modifiée – issu de la définition de "mode de fonctionnement", notes ajoutées]

3.2.29

fréquence moyenne de défaillance dangereuse par heure

PFH* ou *PFH_D

fréquence moyenne de défaillance dangereuse d'un SCS pour réaliser la fonction de sécurité spécifiée pendant une période donnée

Note 1 à l'article: Les deux termes *PFH* et *PFH_D* correspondent à la probabilité de défaillances dangereuses par heure (IEC 62061:2005+AMD1:2012+AMD2:2015).

Note 2 à l'article: Le terme "probabilité moyenne de défaillance dangereuse par heure" n'est plus utilisé dans la présente édition mais l'acronyme *PFH* a été conservé et signifie, lorsqu'il est employé, "fréquence moyenne de défaillance dangereuse [h]".

Note 3 à l'article: L'abréviation "*PFH*" est dérivée du terme anglais développé correspondant "*average frequency of a dangerous failure per hour*".

[SOURCE: IEC 61508-4:2010, 3.6.19, modifiée – terminologie adaptée aux machines, notes existantes supprimées, de nouvelles notes ajoutées]

3.2.30

probabilité de défaillance dangereuse en cas de sollicitation

PFD

indisponibilité de sécurité (voir l'IEC 60050-192) d'un SCS pour réaliser la fonction de sécurité spécifiée sur sollicitation de la part de la machine ou du système de commande de la machine

Note 1 à l'article: L'indisponibilité [instantanée] (selon l'IEC 60050-192) est la probabilité qu'une entité ne soit pas en mesure de réaliser une fonction exigée dans des conditions données à un moment donné, en supposant l'existence des ressources externes requises. Elle est généralement exprimée par $U(t)$.

Note 2 à l'article: La disponibilité [instantanée] ne dépend pas des états (en fonctionnement ou en défaillance) dans lesquels l'article se trouve avant t . Elle caractérise une entité qui doit uniquement être apte à fonctionner lorsqu'elle doit le faire, par exemple, un SCS fonctionnant en mode faible sollicitation.

Note 3 à l'article: Si elle est régulièrement soumise à l'essai, la *PFD* d'un SCS est, par rapport à la fonction de sécurité spécifiée, représentée par une courbe en dents de scie avec une large gamme de probabilités comprises entre un niveau minimal, juste après un essai, et un niveau maximal, juste avant un essai.

Note 4 à l'article: L'abréviation "*PFD*" est dérivée du terme anglais développé correspondant "*probability of dangerous failure on demand*".

[SOURCE: IEC 61508-4:2010, 3.6.17, modifiée – terminologie adaptée aux machines]

3.2.31

probabilité moyenne de défaillance dangereuse en cas de sollicitation

PFD_{avg}

indisponibilité moyenne (voir l'IEC 60050-192) d'un SCS pour réaliser la fonction de sécurité spécifiée sur sollicitation de la part de la machine ou du système de commande de la machine en tant que moyenne dans le temps

Note 1 à l'article: L'indisponibilité moyenne sur un intervalle de temps donné $[t_1, t_2]$ est généralement exprimée par $U(t_1, t_2)$.

Note 2 à l'article: Deux types de défaillances contribuent à la *PFD* et la *PFD_{avg}*: les défaillances dangereuses non détectées qui se sont produites depuis le dernier essai périodique, et les défaillances de sollicitation réelles engendrées par les sollicitations (essais périodiques et sollicitations de sécurité) elles-mêmes. Les premières

défaillances dépendent du temps et sont caractérisées par leur taux de défaillance dangereuse $\lambda_{DU}(t)$, alors que les secondes défaillances ne dépendent que du nombre de sollicitations et sont caractérisées par une probabilité de défaillance par sollicitation (exprimée par γ).

Note 3 à l'article: Dans la mesure où les défaillances sur sollicitation réelles ne peuvent pas être détectées par des essais, il est nécessaire de les identifier et de les prendre en considération lors du calcul des objectifs chiffrés de défaillance.

[SOURCE: IEC 61508-4:2010, 3.6.18, modifiée – terminologie adaptée aux machines]

3.2.32

objectif chiffré de défaillance

PFH ou PDF_{avg} prévisionnel à réaliser pour satisfaire à une ou plusieurs exigences particulières d'intégrité de sécurité

Note 1 à l'article: L'objectif chiffré de défaillance est spécifié en ce qui concerne:

- la probabilité moyenne de défaillance dangereuse de la fonction de sécurité sur sollicitation (pour un mode de fonctionnement faible sollicitation);
- la fréquence moyenne de défaillance dangereuse [h^{-1}] (pour un mode de fonctionnement sollicitation élevée ou un mode de fonctionnement continu).

[SOURCE: IEC 61508-4:2010, 3.5.17, modifiée – expression "probabilité prévisionnelle d'un mode de défaillance dangereux" remplacée par " PFH ou PDF_{avg} prévisionnel", liste à puces déplacée vers la note 1, note existante supprimée]

3.2.33

anomalie

condition anormale qui peut entraîner une réduction de capacité ou la perte de capacité d'un SCS, d'un sous-système ou d'un élément de sous-système à accomplir une fonction exigée

Note 1 à l'article: Dans l'IEC 60050-192, 192-04-01, le défaut d'une entité est décrit comme étant l'inaptitude à fonctionner tel qu'exigé, due à un état interne.

[SOURCE: IEC 61508-4:2010, 3.6.1, modifiée – terminologie adaptée aux machines, note abrégée]

3.2.34

tolérance aux anomalies

aptitude d'un SCS, d'un sous-système ou d'un élément de sous-système à continuer d'accomplir une fonction exigée en présence d'anomalies ou de défaillances

[SOURCE: IEC 61508-4:2010, 3.6.3, modifiée – terminologie adaptée aux machines]

3.2.35

tolérance aux anomalies du matériel

HFT

aptitude d'un sous-système à la perte potentielle de la fonction de sécurité en cas de $N+1$ anomalies au moins

Note 1 à l'article: Une tolérance aux anomalies du matériel N signifie que $N+1$ anomalies d'un sous-système peuvent provoquer la perte de la fonction de sécurité.

Note 2 à l'article: L'abréviation "HFT" est dérivée du terme anglais développé correspondant "*hardware fault tolerance*".

[SOURCE: IEC 61508-2:2010, dérivé de 7.4.4.1.1]

3.2.36

sous-fonction

partie d'une fonction de sécurité dont la défaillance peut donner lieu à une défaillance de la fonction de sécurité

Note 1 à l'article: Dans le présent document, une fonction de sécurité peut être considérée comme un élément AND logique des sous-fonctions.

3.2.37

durée moyenne de fonctionnement avant défaillance

MTTF

espérance mathématique de la durée moyenne avant défaillance

Note 1 à l'article: La *MTTF* représente normalement une valeur moyenne de l'espérance mathématique de la durée de fonctionnement avant défaillance.

Note 2 à l'article: L'abréviation "*MTTF*" est dérivée du terme anglais développé correspondant "*mean time to failure*".

[SOURCE: IEC 60050-192, 192-05-11, modifiée – nouvelle note ajoutée notes initiales supprimées]

3.2.38

durée moyenne de fonctionnement avant défaillance dangereuse

MTTF_D

espérance mathématique de la durée moyenne avant défaillance dangereuse

[SOURCE: Définition dérivée de l'IEC 60050-192, 192-05-11, modifiée – définition limitée aux défaillances dangereuses]

3.2.39

durée moyenne de rétablissement

MTTR

durée prévue jusqu'à la restauration après qu'une anomalie a été produite dans une fonction de sécurité

Note 1 à l'article: La *MTTR* comprend:

- le temps de détection de la défaillance (a); et,
- le temps écoulé avant de commencer la réparation (b); et,
- le temps de réparation effectif (c); et,
- le temps écoulé avant la remise en fonctionnement du composant (d).

L'heure de début de (b) est la fin de (a); l'heure de début de (c) est la fin de (b); l'heure de début de (d) est la fin de (c).

Note 2 à l'article: Pendant cette durée, la machine peut continuer de fonctionner.

Note 3 à l'article: L'abréviation "*MTTR*" est dérivée du terme anglais développé correspondant "*mean time to restoration*".

[SOURCE: IEC 61508-4:2010, 3.6.21, modifiée – terminologie adaptée aux machines et définition plus détaillée]

3.2.40

durée moyenne de réparation

MRT

durée moyenne de réparation après qu'une anomalie a été détectée dans une fonction de sécurité, la machine continuant de fonctionner

Note 1 à l'article: La *MRT* comprend:

- le temps écoulé avant de commencer la réparation (b); et,
- le temps de réparation effectif (c); et,
- le temps écoulé avant la remise en fonctionnement du composant (d).

Note 2 à l'article: Selon le type d'anomalie détecté et la réaction à l'anomalie, les valeurs numériques de *MRT* et de *MTTR* peuvent être différentes.

Note 3 à l'article: L'abréviation "MRT" est dérivée du terme anglais développé correspondant "*mean repair time*".

[SOURCE: IEC 61508-4:2010, 3.6.22, modifiée – terminologie adaptée aux machines et définition plus détaillée, note 1 rendue similaire à 3.2.39, note 2 ajoutée]

3.2.41

temps de sécurité du processus

durée entre l'occurrence d'une défaillance, avec potentialité de donner lieu à un événement dangereux, se produisant dans la machine ou son système de commande et le temps nécessaire pour accomplir l'action dans la machine pour empêcher l'occurrence de l'événement dangereux

Note 1 à l'article: Il est prévu que la fonction de sécurité détecte la défaillance et exécute son action suffisamment tôt pour éviter que l'événement dangereux ne tienne compte d'un décalage de processus (temps d'arrêt, par exemple).

[SOURCE: IEC 61508-4:2010, 3.6.20 – terminologie adaptée aux machines, note 1 ajoutée]

3.2.42

durée de fonctionnement utile

temps minimum écoulé entre l'installation du SCS, du sous-système ou de l'élément de sous-système et le moment auquel les taux de défaillance des composants du SCS, du sous-système ou de l'élément de sous-système ne peuvent plus être prévus, quelle que soit l'exactitude

Note 1 à l'article: En règle générale, cette durée est de 20 ans au maximum, sauf si les fabricants du SCS et de ses sous-systèmes peuvent justifier une durée de vie plus longue en apportant la preuve, reposant sur des calculs, de la validité des données de fiabilité pendant une durée de vie plus longue.

[SOURCE: IEC 61131-6:2012, 3,57, modifiée – terminologie adaptée aux machines, note 1 ajoutée, exemple supprimé]

3.2.43

composant éprouvé

pour une application relative à la sécurité, composant qui a été

- a) largement utilisé dans le passé avec des résultats satisfaisants dans des applications relatives à la sécurité similaires, selon les composants éprouvés indiqués dans les annexes informatives de l'ISO 13849-2, ou
- b) réalisé et vérifié selon des principes qui démontrent son aptitude et sa fiabilité pour des applications relatives à la sécurité

Note 1 à l'article: L'ISO 13849-2 répertorie un éventail de composants et les conditions pour des technologies spécifiques dans lesquelles le composant peut être considéré comme éprouvé.

Note 2 à l'article: Les composants qui viennent d'être développés peuvent être considérés comme étant équivalents aux composants "éprouvés" s'ils satisfont aux conditions de b).

Note 3 à l'article: La décision d'accepter un composant particulier comme étant "éprouvé" dépend de l'application (des influences environnementales, par exemple) et peut être impactée par le produit ou les modifications apportées par le fabricant.

Note 4 à l'article: Les composants électroniques complexes (PLC, microprocesseur, circuit intégré spécifique à l'application, par exemple) ne peuvent pas être considérés comme "éprouvés".

Note 5 à l'article: Un composant éprouvé ne l'est pas en utilisation.

3.2.44

principes de sécurité éprouvés

principes qui se sont révélés efficaces lors de la conception ou de l'intégration des systèmes de commande relatifs à la sécurité dans le passé, afin d'éviter ou de contrôler les anomalies ou défaillances critiques qui peuvent avoir un impact sur les performances d'une fonction de sécurité

Note 1 à l'article: Les principes de sécurité qui viennent d'être développés peuvent être considérés comme étant "éprouvés" s'ils sont vérifiés à l'aide des principes dont la pertinence et la fiabilité ont été démontrées pour des applications relatives à la sécurité.

Note 2 à l'article: Les principes de sécurité éprouvés sont efficaces non seulement contre les défaillances aléatoires du matériel, mais également contre les défaillances systématiques qui peuvent s'insinuer à un certain stade du cycle de vie du produit (les anomalies se produisant lors de la conception, de l'intégration, de la modification ou de la détérioration du produit, par exemple).

Note 3 à l'article: Le Tableau A.2, le Tableau B.2, le Tableau C.2 et le Tableau D.2 des annexes informatives de l'ISO 13849-2:2012 portent sur les principes de sécurité éprouvés pour différentes technologies.

[SOURCE: Définition dérivée de l'ISO 13849-1:2015]

3.2.45

architecture

configuration spécifique des éléments matériels et logiciels dans un SCS

[SOURCE: IEC 61508-4:2010, 3.3.4, modifiée – terminologie adaptée aux machines]

3.2.46

contrainte architecturale

ensemble des exigences d'architecture qui limitent le SIL pouvant être revendiqué pour un sous-système

3.2.47

essai périodique

essai périodique qui peut détecter les anomalies dangereuses non détectées et la dégradation d'un SCS et de ses sous-systèmes de sorte que, si nécessaire, les parties concernées du SCS et de ses sous-systèmes puissent être remises dans un état "comme neuf" ou dans un état aussi proche que possible de celui-ci

Note 1 à l'article: Un essai périodique est prévu afin de confirmer que les parties concernées du SCS sont en état d'assurer l'intégrité de sécurité spécifiée.

Note 2 à l'article: L'efficacité de l'essai périodique dépend de la couverture d'anomalie et de l'efficacité des réparations. Dans la pratique, 100 % de détection des dégradations pouvant donner lieu à des défaillances dangereuses cachées ultérieures n'est pas une valeur aisément atteinte. Pour des éléments complexes ou des caractéristiques de sécurité qui sont difficiles à vérifier, une couverture périodique d'essai de 100 % ne peut en général pas être aisément obtenue.

[SOURCE: IEC 61508-4:2010, 3.8.5, modifiée – terminologie adaptée aux machines, notes 1, 3, 4 supprimées, une nouvelle note 1 ajoutée, note 2 abrégée]

3.2.48

couverture périodique d'essai

terme attribué au pourcentage de défaillances dangereuses non détectées qui sont détectées par une procédure d'essai périodique définie

Note 1 à l'article: Il mesure l'efficacité d'un essai périodique et s'étend de 0 % à 100 % (essai périodique parfait).

Note 2 à l'article: Par exemple, une couverture périodique d'essai de 95 % indique que 95 % de toutes les défaillances possibles non détectées sont détectées pendant l'essai périodique. Elle n'inclut pas le vieillissement ou la dégradation non directement liée à la défaillance de la fonction de sécurité.

Note 3 à l'article: La couverture périodique d'essai peut être calculée au moyen d'une analyse des modes de défaillance et de leurs effets (AMDE) avec le concours d'un avis technique fondé sur des preuves solides.

3.2.49

couverture du diagnostic

DC

proportion de défaillances dangereuses détectées par les essais de diagnostic en ligne automatiques

Note 1 à l'article: La proportion de défaillances dangereuses est calculée en divisant les taux de défaillance dangereuse associés aux défaillances dangereuses détectées par le taux total des défaillances dangereuses.

Note 2 à l'article: La couverture du diagnostic de défaillance dangereuse est calculée selon l'équation suivante, dans laquelle DC est la couverture du diagnostic, λ_{DD} est le taux de défaillance dangereuse détectée et λ_{Dtotal} est le taux total de défaillance dangereuse:

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

Note 3 à l'article: Cette définition s'applique sous réserve que chacun des composants présente des taux de défaillance constants.

Note 4 à l'article: L'abréviation "DC" est dérivée du terme anglais développé correspondant "*diagnostic coverage*".

[SOURCE: IEC 61508-4:2010, 3.8.6, modifiée – une partie de la définition est convertie en une note à l'article]

3.2.50

intervalle entre essais de diagnostic

intervalle de temps entre les essais en ligne qui permettent de détecter les anomalies d'un sous-système, dont la couverture du diagnostic est spécifiée

[SOURCE: IEC 61508-4:2010, 3.8.7, modifiée – remplacement de système relatif à la sécurité par sous-système]

3.2.51

défaillance

cessation de l'aptitude d'une entité (SCS, un sous-système ou un élément de sous-système) à accomplir une fonction exigée

Note 1 à l'article: Les défaillances sont soit aléatoires (pour le matériel), soit systématiques (pour le matériel ou le logiciel).

Note 2 à l'article: Après défaillance, l'entité a une anomalie.

Note 3 à l'article: Une défaillance est un événement, par opposition à une anomalie qui est un état.

Note 4 à l'article: La notion de défaillance, telle qu'elle est définie, ne s'applique pas aux entités constituées seulement de logiciel.

[SOURCE: IEC 61508-4:2010, 3.6.4, modifiée et ISO 12100-1:2010, 3.32]

3.2.52

défaillance dangereuse

défaillance d'un SCS, d'un sous-système ou d'un élément de système ayant une influence sur la mise en œuvre de la fonction de sécurité qui:

- a) empêche le fonctionnement nécessaire de la fonction de sécurité (mode de sollicitation) ou provoque la défaillance d'une fonction de sécurité (mode continu) de sorte que la machine est mise dans un état dangereux ou potentiellement dangereux; ou
- b) diminue la probabilité que la fonction de sécurité fonctionne correctement lorsque c'est nécessaire

[SOURCE: IEC 61508-4:2010, 3.6.4, modifiée – terminologie adaptée aux machines et figure remplacée par une description textuelle et ISO 12100-1:2010, 3.34]

3.2.53

défaillance en sécurité

défaillance d'un SCS, d'un sous-système ou d'un élément de système ayant une influence sur la mise en œuvre de la fonction de sécurité qui:

- a) empêche le fonctionnement nécessaire de la fonction de sécurité avec la potentialité de mettre la machine (ou une partie de celle-ci) dans un état de sécurité ou de maintenir un état de sécurité; ou
- b) augmente la probabilité du fonctionnement parasite de la fonction de sécurité avec potentialité de mettre la machine (ou une partie de celle-ci) dans un état de sécurité ou de maintenir un état de sécurité

[SOURCE: IEC 61508-4:2010, 3.6.8, modifiée – terminologie adaptée aux machines]

3.2.54

proportion de défaillances en sécurité

SFF

proportion du taux global des défaillances d'un sous-système qui n'entraîne pas une défaillance dangereuse

Note 1 à l'article: La couverture du diagnostic (si nécessaire) de chaque sous-système du SCS est prise en compte dans le calcul de la probabilité de défaillance aléatoire du matériel. La proportion de défaillances en sécurité est prise en compte lors de la détermination des contraintes architecturales sur l'intégrité de sécurité du matériel (voir 7.4).

Note 2 à l'article: Les "défaillances sans effet" et les "défaillances partielles" (voir l'IEC 61508-4) ne sont pas utilisées pour calculer la SFF.

Note 3 à l'article: L'abréviation "SFF" est dérivée du terme anglais développé correspondant "*safe failure fraction*".

3.2.55

rapport de défaillance dangereuse

RDF

proportion du taux global des défaillances d'un élément qui peut entraîner une défaillance dangereuse

Note 1 à l'article: L'abréviation "RDF" est dérivée du terme anglais développé correspondant "*ratio of dangerous failure*".

3.2.56

défaillance de cause commune

CCF

défaillance résultant d'un ou de plusieurs événements qui, provoquant des défaillances simultanées de deux ou plusieurs canaux séparés dans un système multicanal, conduit à la défaillance d'une fonction de sécurité

Note 1 à l'article: L'abréviation "CCF" est dérivée du terme anglais développé correspondant "*common cause failure*".

[SOURCE: IEC 61508-4:2010, 3.6.10, modifiée – l'expression "défaillance du système" remplacée par "défaillance d'une fonction de sécurité"]

3.2.57

défaillance aléatoire du matériel

défaillance survenant de manière aléatoire et résultant d'un ou de plusieurs mécanismes de dégradation potentiels au sein du matériel

[SOURCE IEC 61508-4:2010, 3.6.5, modifiée – notes supprimées]

3.2.58

défaillance systématique

défaillance liée de façon déterministe à une certaine cause, ne pouvant être éliminée que par une modification de la conception ou du processus de fabrication, des procédures d'exploitation, de la documentation ou d'autres facteurs appropriés

Note 1 à l'article: Une maintenance corrective sans modification n'élimine généralement pas la cause de la défaillance.

Note 2 à l'article: Une défaillance systématique peut être induite en simulant la cause de la défaillance.

Note 3 à l'article: Parmi les exemples de causes de défaillances systématiques figurent les erreurs humaines dans:

- la spécification des exigences de sécurité;
- la conception, fabrication, installation et/ou exploitation du matériel;
- la conception et/ou mise en œuvre du logiciel.

[SOURCE IEC 61508-4:2010, 3.6.6, modifiée – note 3 légèrement modifiée, note 4 supprimée]

3.2.59

logiciel d'application

logiciel spécifique à l'application qui a été réalisé par le concepteur du SCS et qui contient généralement des séquences logiques, des termes et expressions logiques qui commandent l'entrée, la sortie, les calculs appropriés et les décisions nécessaires pour satisfaire aux exigences fonctionnelles du SCS

3.2.60

logiciel intégré logiciel embarqué

logiciel, fourni comme partie intégrante d'un sous-système type, qui n'est pas destiné à être modifié et qui est lié au fonctionnement du SCS ou du sous-système, et aux services qu'il fournit, par opposition au logiciel d'application

Note 1 à l'article: Le micrologiciel et le logiciel système sont des exemples de logiciel intégré.

3.2.61

langage de variabilité totale

FVL

type de langage qui fournit la possibilité de mettre en œuvre une gamme étendue de fonctions et d'applications

Note 1 à l'article: Un exemple typique de système utilisant le FVL est l'ordinateur d'usage général.

Note 2 à l'article: Le FVL se trouve normalement dans les logiciels intégrés et rarement dans les logiciels d'application.

Note 3 à l'article: Des exemples de FVL incluent: l'Ada, le C, le Pascal, une liste d'instructions, les langages d'assemblage, le C++, le Java, le SQL.

Note 4 à l'article: L'abréviation "FVL" est dérivée du terme anglais développé correspondant "*full variability language*".

[SOURCE: IEC 61511-1:2016, 3.2.75.3, modifiée – la première partie de la définition est supprimée et le rapport au secteur des industries de transformation supprimé]

3.2.62

langage de variabilité limitée

LVL

type de langage qui fournit la possibilité de combiner des fonctions de bibliothèque, prédéfinies, spécifiques à une application, pour mettre en œuvre les spécifications des exigences concernant la sécurité

Note 1 à l'article: Un LVL fournit une correspondance fonctionnelle étroite avec les fonctions exigées pour réaliser l'application.

Note 2 à l'article: Des exemples typiques de LVL sont donnés dans l'IEC 61131-3. Ils comprennent: le langage à contacts, le langage en blocs fonctionnels et le schéma fonctionnel en séquence. Les listes d'instructions et un texte structuré ne sont pas considérés comme LVL.

Note 3 à l'article: Exemple typique de systèmes utilisant le LVL: contrôleur logique programmable (PLC) configuré pour le contrôle de machine

[SOURCE: IEC 61511-1:2016, 3.2.75.2, modifiée – note 1 transformée en définition, note 2 supprimée, note 3 remplacée]

3.2.63**logiciel relatif à la sécurité**

logiciel utilisé pour accomplir des fonctions de sécurité dans un système relatif à la sécurité

3.2.64**vérification**

confirmation, par examen (par exemple essais, analyses) que le SCS, ses sous-systèmes ou éléments de sous-systèmes satisfont aux exigences déterminées par la spécification appropriée

EXEMPLE: Les activités de vérification incluent:

- les revues relatives aux sorties d'une phase (documents concernant toutes les phases) destinées à assurer la conformité aux objectifs et exigences de la phase, et prenant en compte les entrées spécifiques à cette phase;
- les revues de conception;
- les essais réalisés sur les produits mis au point afin d'assurer que leur fonctionnement est conforme à leur spécification;
- les essais d'intégration réalisés lors de l'assemblage élément par élément de différentes parties d'un système, à partir d'essais d'environnement, afin d'assurer que toutes les parties fonctionnent les unes avec les autres conformément aux spécifications.

[SOURCE: IEC 61508-4:2010, 3.8.1, modifiée – terminologie adaptée aux machines, note supprimée]

3.2.65**validation (de la fonction de sécurité)**

confirmation par examen (par exemple essais, analyses) que le SCS satisfait aux exigences de sécurité fonctionnelle pour une application spécifique

[SOURCE: IEC 61508-4:2010, 3.8.2, modifiée – terminologie adaptée aux machines, notes supprimées]

3.2.66**gestion de configuration**

discipline d'identification des composants d'un système évolutif ayant pour objectif de maîtriser les modifications de ces composants et de maintenir la continuité et la traçabilité tout au long du cycle de vie

[SOURCE IEC 61508-4:2010, 3.7.3, modifiée – note supprimée]

3.2.67**référentiel (de configuration)**

ensemble précis d'éléments (matériel, logiciels, documentation, essais, etc.) d'un SCS à un instant donné

Note 1 à l'article: Un référentiel sert de base pour la vérification, la validation, la modification et les changements.

Note 2 à l'article: Si un élément est modifié, le statut du référentiel est intermédiaire jusqu'à ce qu'un nouveau référentiel ait été défini.

3.2.68**état de sécurité**

état de la machine lorsque la sécurité est réalisée

Note 1 à l'article: L'état de sécurité n'inclut pas le rétablissement des défaillances initiales du matériel.

[SOURCE: IEC 61508-4:2010, 3.1.13, modifiée – terminologie adaptée aux machines, note initiale supprimée, note 1 ajoutée]

3.2.69 sécurité

- 1) mesures prises pour protéger un système
- 2) état d'un système qui résulte de l'établissement et du maintien de mesures visant à le protéger
- 3) état des ressources système protégées contre les accès non autorisés ou contre la modification, la destruction ou la perte non autorisée ou accidentelle
- 4) aptitude d'un système informatique à garantir que des personnes et systèmes non autorisés ne peuvent ni modifier les logiciels et ses données ni accéder aux fonctions système, et à assurer que l'accès n'est pas refusé aux personnes et systèmes autorisés
- 5) prévention de toute pénétration illégale ou indésirable ou gêne du bon fonctionnement prévu d'un système d'automatisation et de commande industrielle

Note 1 à l'article: Les mesures peuvent être des contrôles liés à la sécurité physique (contrôle des accès physiques aux actifs informatiques) ou à la sécurité logique (capacité de connexion à un système et une application donnés).

[SOURCE: Cette source n'existe que dans la langue anglaise]

3.3 Abréviations

Le Tableau 2 présente les abréviations utilisées dans le présent document.

Tableau 2 – Abréviations utilisées dans l'IEC 62061

CCF	Common Cause Failures (défaillances de cause commune)
DC	Diagnostic Coverage (couverture du diagnostic)
CEM	Compatibilité électromagnétique
FVL	Full Variability Language (langage de variabilité totale)
E/S	Entrée/Sortie
LVL	Limited Variability Language (langage de variabilité limitée)
HFT	Hardware Fault Tolerance (tolérance aux anomalies du matériel)
HW	Hardware (matériel)
PFH, PFH_D	average frequency of dangerous failure per Hour (fréquence moyenne de défaillance dangereuse par heure)
MRT	Mean Repair Time (durée moyenne de réparation)
MTTF	Mean Time to Failure (durée moyenne de fonctionnement avant défaillance)
$MTTF_D$	Mean Time to Dangerous Failure (durée moyenne de fonctionnement avant défaillance dangereuse)
MTTR	Mean Time To Restoration (durée moyenne de rétablissement)
PFD	probability of dangerous failure on demand (probabilité de défaillance dangereuse en cas de sollicitation)
PFD_{avg}	average probability of dangerous failure on demand (probabilité moyenne de défaillance dangereuse en cas de sollicitation)
PL	Performance Level (niveau de performance)
PLC	Programmable Logic Controller (contrôleur logique programmable)
RDF	Ratio of Dangerous Failure (rapport de défaillance dangereuse)
SFF	Safe Failure Fraction (proportion de défaillances en sécurité)
SIL	Safety Integrity Level (niveau d'intégrité de sécurité)
SCS	Safety-related Control System (système de commande relatif à la sécurité)
SRS	Safety Requirements Specification (spécification des exigences de sécurité)
SW	Software (logiciel)

4 Processus de conception d'un SCS et gestion de la sécurité fonctionnelle

4.1 Objectifs

L'Article 4 a pour objet de décrire le processus de conception et les tâches qui doivent être exécutés pour réaliser chaque fonction de sécurité effectuée par la partie concernée du système de commande d'une machine donnée.

4.2 Processus de conception

Si un besoin de réduction du risque a été identifié après une appréciation du risque dont a fait l'objet l'ensemble de la machine conformément à l'ISO 12100 (voir la Figure 2), et si certaines mesures de réduction du risque choisies dépendent du système de commande, les fonctions de sécurité correspondantes doivent être spécifiées.

NOTE 1 Des exemples de fonctions de sécurité sont donnés à l'Annexe H.

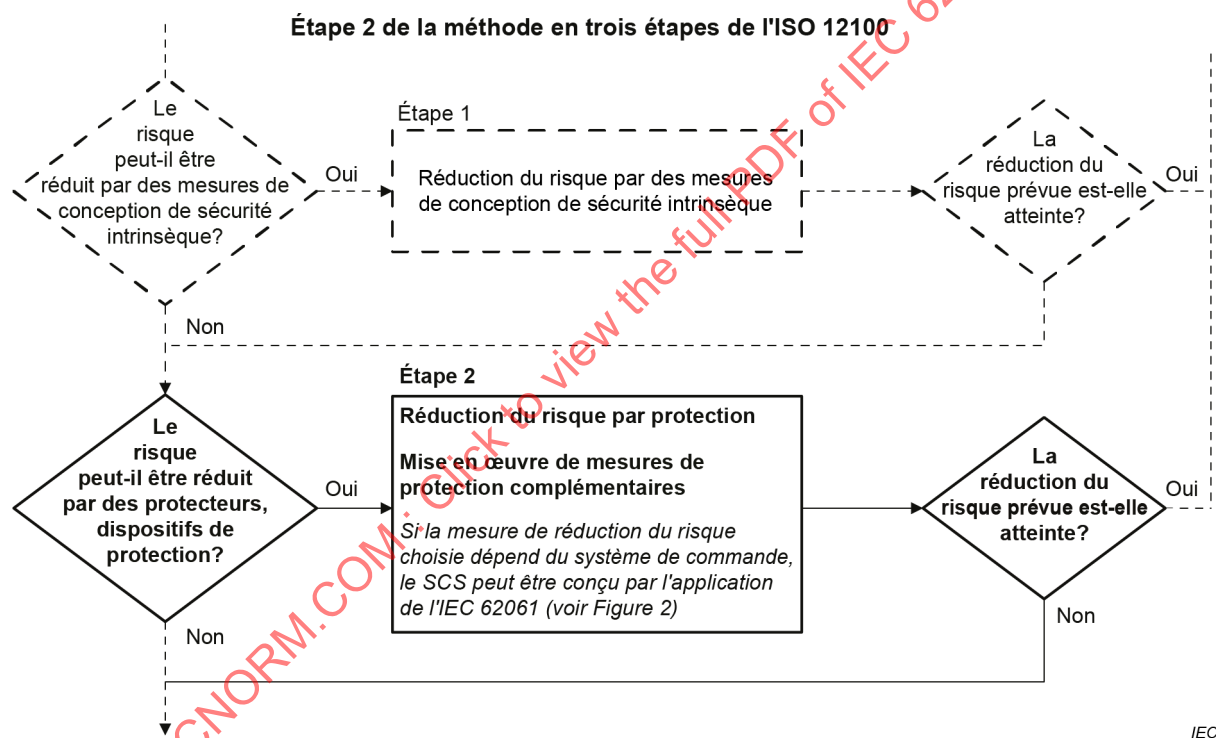
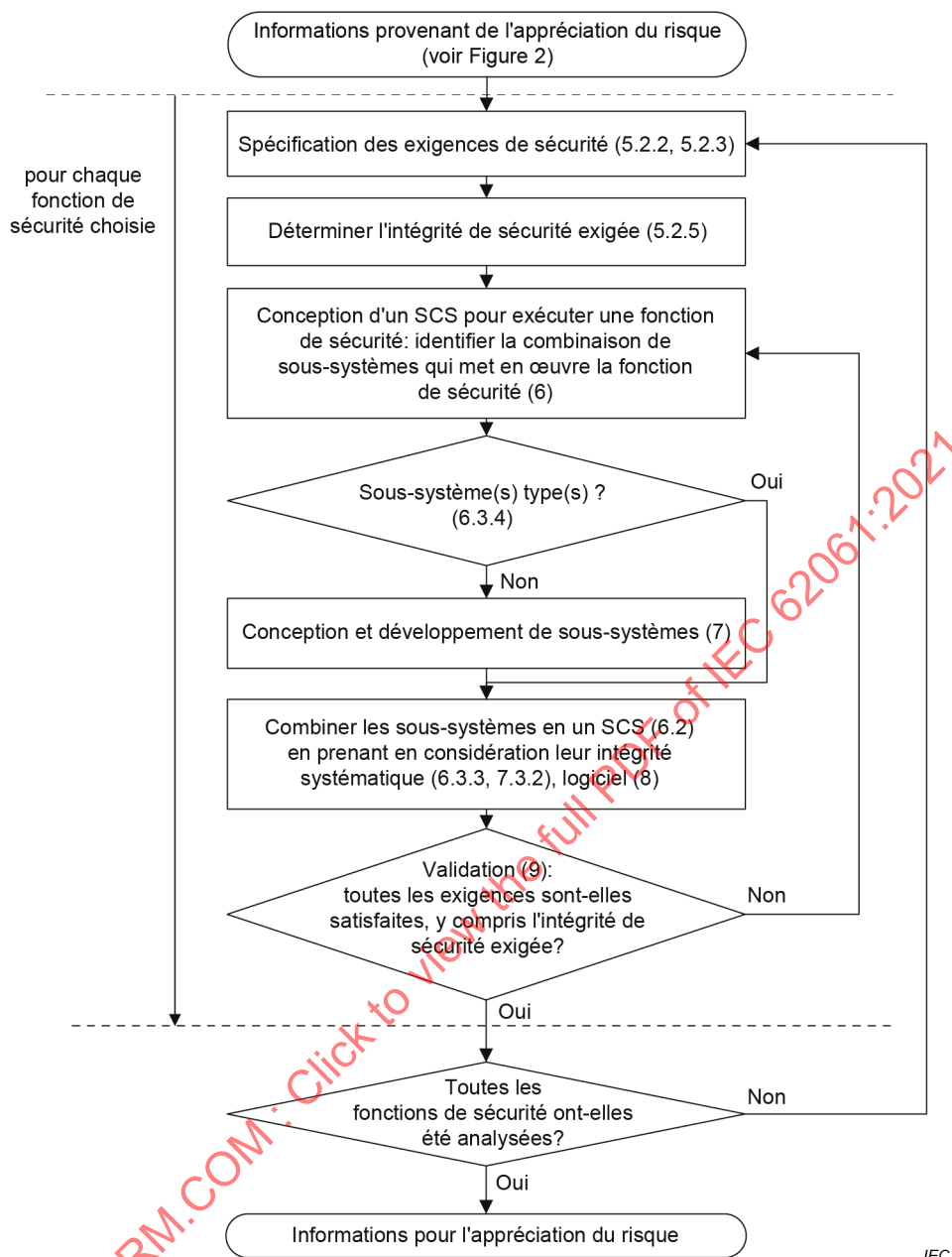


Figure 2 – Intégration dans le processus de réduction du risque de l'ISO 12100 (extrait)

NOTE 2 La Figure 2 représente dans quelle mesure le SCS contribue au processus de réduction du risque de l'ISO 12100: Étape 2: Le SCS prend en charge les mesures de prévention combinées par la mise en œuvre de fonctions de sécurité. L'ISO 12100 fournit également les règles générales de conception de la machine applicables pour la conception du SCS (voir 6.2.11 et 6.2.12 de l'ISO 12100:2010).

Le processus de conception (voir la Figure 3) de chaque fonction de sécurité mise en œuvre par un système de commande relatif à la sécurité (SCS) doit au moins inclure la spécification de fonction de sécurité (voir l'Article 5), la conception du système de commande relatif à la sécurité (voir l'Article 6) et les activités connexes de vérification et de validation.



NOTE Chaque étape décrite dans le schéma de procédé inclut également les activités de vérification.

Figure 3 – Processus itératif de conception du système de commande relatif à la sécurité

La fonction de sécurité suivant l'intégrité de sécurité exigée déterminée doit être réalisée

- à l'aide d'un SCS déjà développé qui satisfait à l'intégrité de sécurité exigée, ou
- en concevant un nouveau SCS à l'aide de sous-systèmes types selon l'Article 6 et/ou en concevant de nouveaux sous-systèmes selon l'Article 7.

Si des considérations supplémentaires en matière de conception de logiciels sont nécessaires, l'Article 8 s'applique.

Une fonction de sécurité peut être mise en œuvre par un ou plusieurs sous-systèmes d'un système de commande relatif à la sécurité (SCS), et plusieurs fonctions de sécurité peuvent partager un ou plusieurs sous-systèmes (une unité logique, un ou plusieurs éléments de commande de puissance, par exemple). Voir la Figure 4. Un système de commande peut être subdivisé en une partie relative à la sécurité et une partie qui ne l'est pas. Il est possible qu'un sous-système soit impliqué tant dans la mise en œuvre des fonctions de sécurité que dans la mise en œuvre des fonctions de commande. Le concepteur peut utiliser l'une des technologies disponibles, indépendamment ou en combinaison.

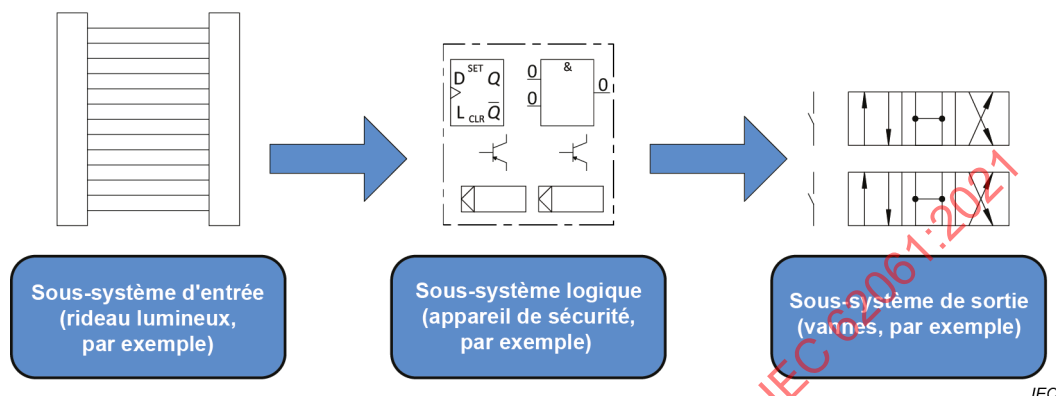


Figure 4 – Exemple de combinaison de sous-systèmes en un SCS

4.3 Gestion de la sécurité fonctionnelle à l'aide d'un plan de sécurité fonctionnelle

Le présent paragraphe spécifie les activités techniques et de gestion nécessaires à la réalisation de la sécurité fonctionnelle exigée du SCS.

NOTE 1 Pour plus d'informations, voir l'IEC 61508-1:2010, Article 6.

Un plan de sécurité fonctionnelle doit être dressé et documenté pour chaque projet de conception de SCS, et doit être mis à jour autant que nécessaire. Le plan de sécurité fonctionnelle est destiné à fournir des mesures de prévention contre toute spécification, mise en œuvre ou modification incorrectes.

Le plan de sécurité fonctionnelle doit identifier les activités pertinentes (voir la Figure 3) et doit être adapté au projet. Voir les exemples à l'Annex I.

NOTE 2 Le plan de sécurité fonctionnelle peut faire partie d'un plan global de conception de machine.

NOTE 3 Le contenu du plan de sécurité fonctionnelle dépend des circonstances particulières, qui peuvent comprendre:

- la dimension du projet;
- le degré de complexité;
- le degré d'innovation de la conception et de la technologie;
- le degré de normalisation des caractéristiques de conception;
- la ou les conséquences possibles en cas de défaillance.

En particulier, le plan de sécurité fonctionnelle doit:

- a) identifier les activités appropriées spécifiées de l'Article 5 à l'Article 9 et le moment auquel elles doivent être réalisées;
- b) décrire la politique et la stratégie pour satisfaire aux exigences de sécurité fonctionnelle spécifiées;

- c) décrire la stratégie pour réaliser la sécurité fonctionnelle pour le logiciel d'application, les résultats d'un développement, l'intégration, la vérification et la validation;
- d) identifier les personnes, services ou autres unités ainsi que les ressources responsables de l'exécution et de la revue de chacune des activités spécifiées de l'Article 5 à l'Article 9.

NOTE 4 Le niveau de compétence approprié des personnes concernées (c'est-à-dire leur formation, leurs connaissances techniques, leur expérience et leurs qualifications) est pris en compte. Le caractère approprié des compétences est pris en considération en relation avec l'application particulière, en tenant compte de tous les facteurs pertinents, y compris:

- a) la responsabilité des personnes;
 - b) le niveau de supervision exigé;
 - c) les conséquences potentielles en cas de défaillance du SCS;
 - d) les niveaux d'intégrité de sécurité du SCS;
 - e) le degré d'innovation de la conception, des procédures de conception ou de l'application;
 - f) l'expérience et sa pertinence pour les obligations spécifiques à respecter et la technologie utilisée;
 - g) le type de compétences appropriées aux circonstances (les qualifications, l'expérience, la formation et la pratique, ainsi que les aptitudes à diriger une équipe et à prendre des décisions, par exemple);
 - h) les connaissances techniques adaptées au domaine d'application et à la technologie;
 - i) les connaissances techniques en matière de sécurité adaptées à la technologie;
 - j) les connaissances du cadre réglementaire et juridique en matière de sécurité;
 - k) la pertinence des qualifications aux activités spécifiques à réaliser.
- e) identifier ou établir les procédures et les ressources d'enregistrement et d'entretien des informations appropriées à la sécurité fonctionnelle d'un SCS;

NOTE 5 Les éléments suivants sont pris en considération:

- les résultats de l'identification des phénomènes dangereux et de l'appréciation du risque;
 - les matériels utilisés pour les fonctions relatives à la sécurité ainsi que leurs exigences concernant la sécurité;
 - l'organisation responsable du maintien de la sécurité fonctionnelle;
 - les procédures nécessaires pour réaliser et maintenir la sécurité fonctionnelle (y compris les modifications des SCS).
- f) décrire la stratégie de gestion de configuration (voir 4.4) prenant en compte les points d'organisation, tels que les personnes autorisées et les structures internes de l'organisation;
 - g) décrire la stratégie de modification (voir 4.5);
 - h) établir un plan de vérification qui doit inclure:
 - des informations sur le moment auquel la vérification doit être réalisée;
 - des informations sur les personnes, services ou unités qui doivent réaliser la vérification;
 - les critères de choix entre les stratégies et techniques de vérification;
 - les critères de choix et d'utilisation des matériels d'essai;
 - les critères de choix des activités de vérification;
 - les critères d'acceptation; et
 - les moyens à utiliser pour l'évaluation des résultats de la vérification;
 - i) établir un plan de validation comprenant:
 - les résultats de la vérification précédente;
 - des informations sur le moment auquel la validation doit être réalisée;
 - l'identification des modes de fonctionnement appropriés de la machine (par exemple fonctionnement normal, réglage);
 - les exigences par rapport auxquelles le SCS doit être validé;
 - la stratégie technique de validation, par exemple les méthodes analytiques ou les essais statistiques;

- les critères d'acceptation; et
- les actions à mener en cas de non-satisfaction aux critères d'acceptation.

NOTE 6 Le plan de validation indique si le SCS et ses sous-systèmes doivent être soumis à des essais individuels de série, des essais de type et/ou des essais sur prélèvement.

4.4 Gestion de configuration

Les principaux aspects opérationnels de la gestion de configuration sont

- l'**identification** de la structure du SCS, qui identifie, par exemple, le système, les sous-systèmes, les fonctions, les blocs fonctionnels, les documents de gestion, les outils de création d'un référentiel;
- le **contrôle** de la publication d'un élément créé à un instant précis de chaque phase du cycle de vie;
- l'**enregistrement** et la **consignation** du statut de chaque élément qui fait et/ou va faire partie d'un référentiel;
- l'**audit** et l'**examen** de tous les éléments et le maintien de la cohérence entre tous les éléments d'un référentiel.

Des procédures doivent être développées pour la gestion de configuration du SCS au cours des phases globales du cycle de vie de sécurité du système SCS et des logiciels, y compris en particulier:

- a) le point, en fonction des phases spécifiques, auquel le contrôle de configuration formel doit être mis en œuvre;
- b) les procédures à utiliser pour identifier sans équivoque toutes les parties constitutives du matériel et des logiciels;
- c) les procédures de prévention de l'entrée des éléments non autorisés dans le service.

Les procédures de gestion de configuration doivent être mises en œuvre selon le plan de sécurité fonctionnelle (voir 4.3).

Les procédures pour un processus de maîtrise des modifications approprié doivent prendre en considération les exigences des procédures de définition d'un référentiel unique de chaque version du SCS.

4.5 Modification

Si une modification doit être mise en œuvre, les activités concernées doivent être identifiées de manière spécifique et un plan d'action doit être élaboré et documenté avant de procéder à une telle modification.

NOTE 1 La demande de modification peut être une conséquence, par exemple:

- d'une modification apportée à la spécification des exigences de sécurité;
- des conditions réelles d'emploi;
- de l'expérience d'un incident/accident;
- d'une modification du matériau traité;
- de l'obsolescence;
- de modifications de la machine ou de ses modes de fonctionnement.

NOTE 2 Les interventions (par exemple réglage, calibrage, réparations) sur le SCS réalisées conformément aux informations pour l'utilisation ou au manuel d'instruction pour le SCS ne sont pas considérées comme étant une modification dans le contexte du présent paragraphe.

La ou les raisons de la demande de modification doivent être documentées.

Les effets de la modification demandée doivent être analysés afin d'établir les conséquences sur la sécurité fonctionnelle.

L'analyse d'impact de la modification et ses conséquences sur la sécurité fonctionnelle du SCS doivent être documentées.

Toutes les modifications acceptées qui ont des conséquences sur le SCS doivent provoquer un retour à une phase de conception appropriée pour son matériel et/ou pour son logiciel (par exemple, spécification, conception, intégration, installation, mise en service et validation). Toutes les phases et procédures de gestion ultérieures doivent alors être effectuées conformément aux procédures spécifiées dans le présent document pour chaque phase particulière. Tous les documents appropriés doivent être révisés, amendés et réédités en conséquence.

5 Spécification d'une fonction de sécurité

5.1 Objectifs

Le présent article établit les procédures de spécification des exigences de la ou des fonctions de sécurité à mettre en œuvre par le SCS.

5.2 Spécification des exigences de sécurité (SRS)

5.2.1 Généralités

Chaque fonction de sécurité doit être spécifiée par:

- la spécification des exigences fonctionnelles (voir 5.2.3);
- la spécification des exigences d'intégrité de sécurité (voir 5.2.5)

et celles-ci doivent être documentées dans la spécification des exigences de sécurité (SRS).

Lorsqu'une norme de produit spécifie les exigences de sécurité pour la conception d'un SCS ou d'un sous-système (l'ISO 13851 pour les dispositifs de commande bimanuelle, par exemple), il convient de les prendre en considération.

5.2.2 Informations à mettre à disposition

Les informations suivantes doivent être utilisées pour produire à la fois la spécification des exigences fonctionnelles et la spécification des exigences d'intégrité de sécurité du SCS:

- les résultats de l'appréciation du risque pour la machine, y compris toutes les fonctions de sécurité déterminées comme nécessaires au processus de réduction du risque pour chaque phénomène dangereux spécifique;
- les caractéristiques de fonctionnement de la machine, y compris:
 - les modes de fonctionnement de la machine,
 - la durée de cycle,
 - le fonctionnement en temps de réponse,
 - les conditions environnementales,
 - l'interaction de personne(s) avec la machine (par exemple, réparation, réglage, nettoyage);

- toutes les informations appropriées pour la ou les fonctions de sécurité qui peuvent avoir une influence sur la conception du SCS, y compris, par exemple:
 - une description du comportement de la machine qu'une fonction de sécurité est destinée à réaliser ou empêcher;
 - toutes les interfaces entre les fonctions de sécurité, et entre les fonctions de sécurité et d'autres fonctions (qu'elles soient à l'intérieur ou à l'extérieur de la machine);
 - les fonctions réactions aux anomalies exigées de la fonction de sécurité.

NOTE Certaines informations peuvent ne pas être disponibles ou suffisamment définies avant le démarrage du processus itératif de conception du SCS. Une mise à jour des spécifications des exigences de sécurité du SCS peut donc être exigée pendant le processus de conception.

5.2.3 Spécification des exigences fonctionnelles

La spécification des exigences fonctionnelles doit décrire les informations concernant chaque fonction de sécurité à réaliser, y compris, le cas échéant:

- une description de chaque fonction de sécurité;
- la ou les conditions (par exemple, mode de fonctionnement) de la machine dans laquelle la fonction de sécurité doit être active, désactivée, configurée ou paramétrée;
- la priorité associée à ces fonctions, lesquelles peuvent être actives simultanément et provoquer une action conflictuelle;
- la réinitialisation d'une fonction de sécurité;
- la fréquence de fonctionnement de chaque fonction de sécurité (taux de cycles de manœuvres, cycles de fonctionnement);
- mode de fonctionnement à sollicitation;

NOTE 1 Pour les définitions, voir 3.2.26, 3.2.27, 3.2.28.

- le temps de réponse exigé de chaque fonction de sécurité;
- l'interface ou les interfaces des fonctions de sécurité avec d'autres fonctions de la machine;

NOTE 2 Ces détails sur l'interface peuvent inclure une description des méthodes destinées à fournir des informations d'état aux utilisateurs de la machine.

- une description de la ou des fonctions réactions aux anomalies et de toutes les contraintes s'exerçant, par exemple, sur le redémarrage ou le fonctionnement continu de la machine, dans les cas où la première réaction à l'anomalie est d'arrêter la machine;
- les essais et les installations associées (par exemple, les équipements d'essai, les ports d'accès d'essai);
- une description de l'environnement de fonctionnement (par exemple, immunité électromagnétique, température, humidité, poussière, substances chimiques, vibration mécanique et choc);

NOTE 3 La spécification de la condition d'environnement électromagnétique relève du domaine d'application de l'IEC 61000-1-2. L'environnement électromagnétique est défini comme l'ensemble des phénomènes électromagnétiques présents en un endroit particulier. Ces phénomènes peuvent varier dans le temps.

Par exemple, l'environnement électromagnétique est influencé par:

- des sources fixes ou mobiles d'énergie électromagnétique,
- un matériel basse, moyenne et haute tension,
- des systèmes de commande, de signalisation, de communication et d'alimentation,
- des éléments rayonnants intentionnels,
- des processus physiques (décharges atmosphériques, actions de commutation, par exemple),
- des transitoires aléatoires ou occasionnels,

qui peuvent tous générer des perturbations qui compromettent le système ou l'élément relatif à la sécurité à l'étude.

- les taux de cycles de manœuvres, le cycle de fonctionnement et/ou la catégorie d'emploi, pour les dispositifs destinés à être utilisés dans la fonction de sécurité;

NOTE 4 Le cycle de fonctionnement des sous-systèmes ou éléments de sous-système peut être plus élevé que celui exigé pour la fonction de sécurité, par exemple lorsqu'il est également utilisé pour des fonctions de machine non relatives à la sécurité (le nombre total de cycles est à prendre en considération).

- d'autres exigences particulières qui peuvent avoir un impact sur la sécurité fonctionnelle.

5.2.4 Estimation du mode de fonctionnement à sollicitation

Le mode de fonctionnement à sollicitation doit être évalué en appliquant les définitions correspondantes. Le présent document se concentre sur le mode sollicitation élevée et le mode continu bien que le mode de fonctionnement à faible sollicitation soit possible pour une fonction de sécurité. Lorsque le taux de sollicitation estimé est faible, un mode à forte sollicitation peut être défini par hypothèse par l'activation de la fonction de sécurité au moins une fois par an. Appliquer ensuite le présent document pour la conception. Il s'agit d'une application pure et simple de la définition, représentée à la Figure 5 comme un flux de travaux.

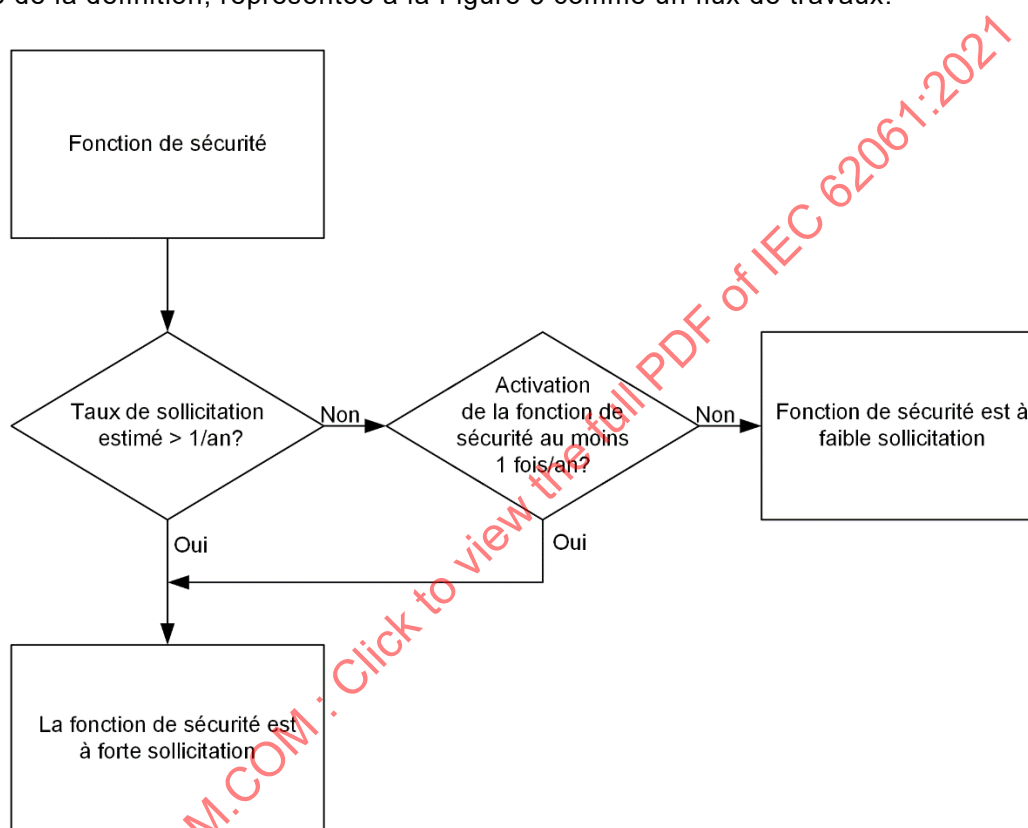


Figure 5 – Définition possible par hypothèse d'un mode à forte sollicitation par l'activation d'une fonction de sécurité à faible sollicitation au moins une fois par an

5.2.5 Spécification des exigences d'intégrité de sécurité

Les exigences d'intégrité de sécurité pour chaque fonction de sécurité doivent être déduites de l'appréciation du risque afin de pouvoir obtenir avec certitude la réduction du risque nécessaire. Dans le présent document, une exigence d'intégrité de sécurité est exprimée par un objectif chiffré de défaillance pour la fréquence moyenne de défaillance dangereuse par heure (*PFH*).

L'intégrité de sécurité exigée pour chaque fonction de sécurité à réaliser par un SCS doit être spécifiée en ce qui concerne le SIL selon le Tableau 3 et doit être documentée.

Tableau 3 – SIL et limites des valeurs de PFH

SIL	Limites des valeurs de PFH (1/h)
1	$< 10^{-5}$
2	$< 10^{-6}$
3	$< 10^{-7}$

La détermination de l'intégrité de sécurité exigée résulte de l'appréciation du risque et se rapporte au niveau de réduction du risque à atteindre par le SCS. Des exemples de méthodologie sont donnés à l'Annexe A.

NOTE 1 Lorsqu'une norme de produit spécifie un SIL exigé pour une fonction de sécurité, celui-ci a priorité par rapport à l'Annexe A.

NOTE 2 D'autres recommandations relatives à la relation entre l'appréciation du risque selon l'ISO 12100 et les normes de produits sont fournies dans l'ISO TR 22100-1.

6 Conception d'un SCS

6.1 Généralités

Le SCS doit être conçu selon la spécification des exigences de sécurité (voir 5.2), à l'aide d'un ou de plusieurs sous-systèmes:

- en choisissant les sous-systèmes (voir 6.2, 6.3 et l'Article 7);
- en déterminant l'intégrité de sécurité (voir 6.4);
- en satisfaisant aux exigences de l'intégrité de sécurité systématique du SCS (voir 6.5), y compris, le cas échéant, l'immunité électromagnétique (voir 6.6), la sécurité (voir 6.8), les essais périodiques (voir 6.9) et les logiciels (voir 6.7 et l'Article 8).

6.2 Architecture de sous-système en fonction de la décomposition descendante

L'Article 6 ci-après décrit le processus de conception d'un SCS. Un SCS peut inclure:

- un ou plusieurs sous-systèmes types, et/ou
- un ou plusieurs sous-systèmes développés selon le présent document, en fonction de l'élément ou des éléments de sous-système (voir l'Article 7).

NOTE 1 Le concepteur d'un sous-système type peut être le fabricant d'une machine ou d'un dispositif.

NOTE 2 Les valeurs caractéristiques pertinentes en matière d'intégrité de sécurité proviennent du concepteur du sous-système type.

6.3 Méthodologie de base – Utilisation du sous-système

6.3.1 Généralités

Chaque fonction de sécurité identifiée dans le processus de réduction du risque (voir l'Article 4) est réalisée par un SCS composé d'au moins un sous-système. La défaillance d'un sous-système se traduit par la perte de l'ensemble de la fonction de sécurité. Le 6.2 décrit le principe de cette tâche d'allocation.

Lorsqu'un SCS ou une partie d'un SCS (c'est-à-dire un ou plusieurs de ses sous-systèmes) doit réaliser à la fois des fonctions de sécurité et d'autres fonctions, l'ensemble de ses matériels et logiciels doit être considéré comme étant relatif à la sécurité, sauf s'il peut être démontré que la mise en œuvre des fonctions de sécurité et des autres fonctions est suffisamment indépendante (c'est-à-dire que le fonctionnement normal ou la défaillance de n'importe laquelle des autres fonctions ne compromet pas les fonctions de sécurité).

NOTE 1 L'indépendance suffisante dans la mise en œuvre est prouvée en démontrant que la probabilité d'une défaillance dépendante entre des parties non relatives à la sécurité et des parties relatives à la sécurité est équivalente à celle du niveau d'intégrité de sécurité du SCS. L'Annexe F de l'IEC 61508-3:2010 décrit les techniques permettant d'assurer la non-interférence entre les éléments logiciels.

Dans le cas d'un SCS ou de ses sous-systèmes mettant en œuvre des fonctions de sécurité de différents niveaux d'intégrité de sécurité, le matériel et le logiciel doivent être considérés comme exigeant le niveau d'intégrité de sécurité le plus élevé, sauf s'il peut être démontré que la mise en œuvre des fonctions de sécurité des différents niveaux d'intégrité de sécurité est suffisamment indépendante.

NOTE 2 L'indépendance suffisante de la mise en œuvre est établie en démontrant que la probabilité d'une défaillance dépendante entre des parties non relatives à la sécurité et des parties relatives à la sécurité est suffisamment faible comparée au niveau d'intégrité de sécurité le plus élevé associé aux fonctions de sécurité concernées.

Lorsque la communication de données numériques est utilisée dans le cadre de la mise en œuvre d'un SCS, elle doit satisfaire aux exigences appropriées de l'IEC 61508-2:2010, 7.4.11 (qui fait référence à l'IEC 61784-3 (toutes les parties) relative aux bus de terrain de sécurité fonctionnelle) selon le ou les niveaux SIL cibles de la ou des fonctions de sécurité.

6.3.2 Décomposition du SCS

Chaque fonction de sécurité doit être décomposée en une structure de sous-fonctions. Le processus de décomposition doit mener à une structure de sous-fonctions qui décrit complètement les exigences fonctionnelles et les exigences d'intégrité du SCS. Il convient d'appliquer ce processus vers le bas jusqu'à un niveau permettant de déterminer les exigences fonctionnelles et les exigences d'intégrité pour chaque sous-fonction à attribuer à un seul sous-système.

La Figure 6 présente des exemples de décompositions classiques commençant par la détection et l'évaluation d'un "événement déclencheur" et se terminant par une sortie provoquant la réaction d'un "actionneur".

Pour chaque sous-fonction, ce qui suit doit être spécifié:

- les exigences de sécurité (fonctionnelles et d'intégrité), et
- les entrées et sorties de chaque sous-fonction.

NOTE 1 Les entrées et sorties de chaque sous-fonction sont les informations qui sont transférées, par exemple la vitesse, la position, le mode de fonctionnement, etc.

NOTE 2 Les sous-fonctions peuvent être associées à des fonctions de diagnostic (voir 7.4.3.3, Couverture du diagnostic).

NOTE 3 Un SCS peut être composé d'un seul sous-système. Un capteur "intelligent" (un lecteur de codes à barres laser, par exemple) avec un dispositif de coupure de sortie intégré (un relais, par exemple) est un exemple de mise en œuvre de SCS avec un seul sous-système.

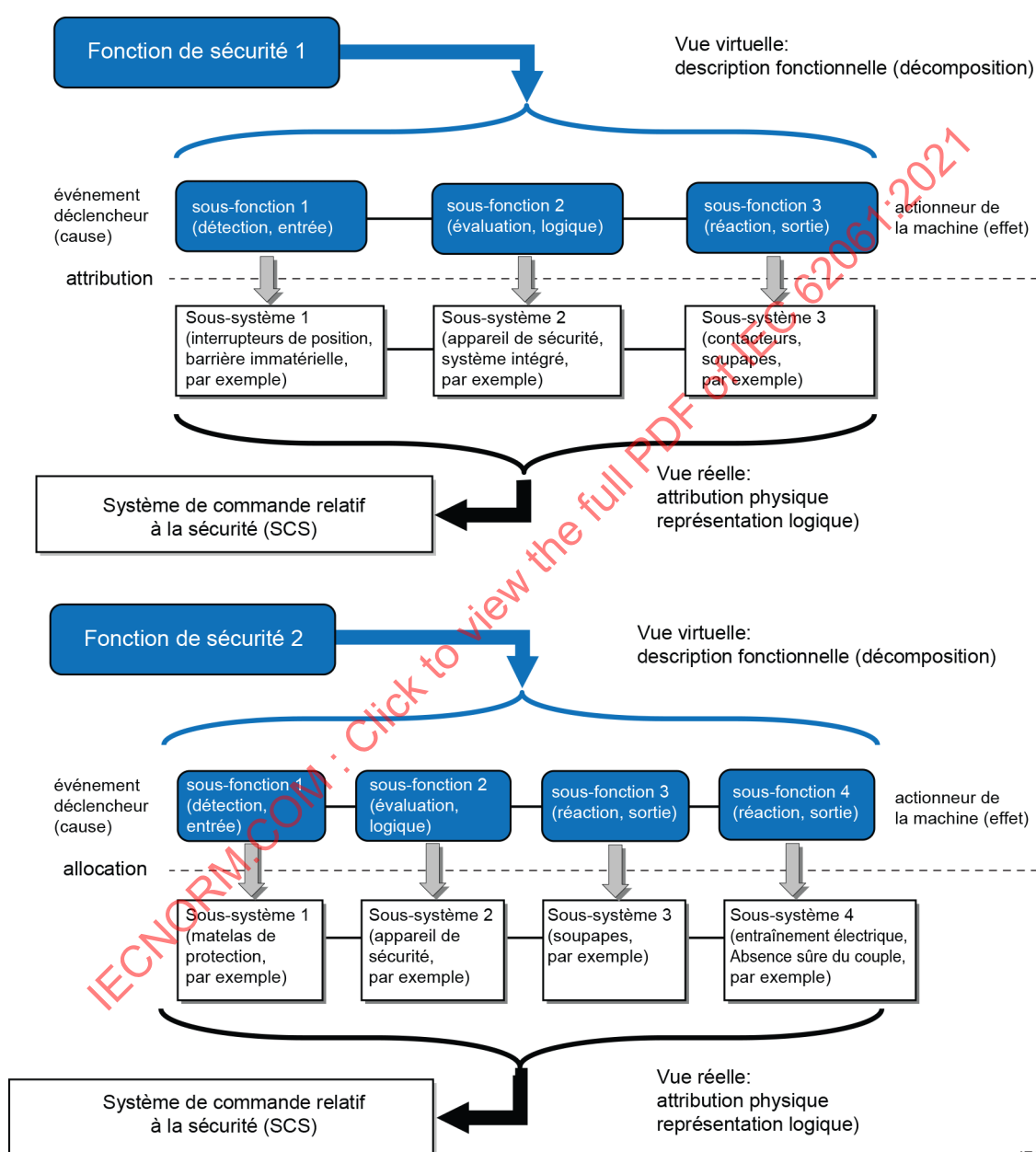
NOTE 4 Un sous-système qui met en œuvre une sous-fonction peut être composé de plusieurs unités physiques. Il s'agit, par exemple, d'un appareil de sécurité équipé d'une unité d'entrée, d'une unité logique et d'une unité de sortie (et d'une communication de bus de terrain relatif à la sécurité) séparées. Le fabricant peut fournir séparément les données relatives à la sécurité pour les unités.

Un autre exemple est un module de relais de sécurité qui surveille l'état du dispositif d'entrée. Si le module de relais de sécurité ne contient pas suffisamment de contacts de sortie pour la sous-fonction spécifique, un module de sécurité d'extension peut être ajouté. Le fabricant fournit séparément les données relatives à la sécurité pour tous les modules.

NOTE 5 Lors de la décomposition des exigences de sécurité en sous-exigences, des processus appropriés de documentation et de gestion de configuration sont menés pour assurer le maintien de la traçabilité bidirectionnelle entre les exigences décomposées.

La décomposition d'un SCS en sous-systèmes (représentée à la Figure 6) est classique, mais l'ensemble du SCS peut être réalisé par un certain nombre de sous-systèmes.

La Figure 6 ne présente pas les fonctions de diagnostic possibles qui peuvent être exigées pour satisfaire aux exigences de sécurité.



IEC

NOTE 1 La communication de bus de terrain peut faire partie d'un ou de plusieurs sous-systèmes.

NOTE 2 Les aspects relatifs à l'interconnexion (le câblage, par exemple) peuvent être pertinents dans un ou plusieurs sous-systèmes (voir 7.3.2.2).

Figure 6 – Exemples de décomposition classique d'une fonction de sécurité en sous-fonctions et de son attribution aux sous-systèmes

6.3.3 Attribution de sous-fonction

Chaque sous-fonction doit être attribuée à un sous-système de l'architecture du SCS. Plusieurs sous-fonctions (la mise en œuvre de différentes fonctions de sécurité, par exemple) peuvent être attribuées à un sous-système.

NOTE Un exemple de sous-système qui met en œuvre plusieurs sous-fonctions est un appareil de sécurité faisant office d'unité logique pour la fonction de protecteur avec dispositif de verrouillage et la fonction de protection contre la survitesse.

6.3.4 Utilisation d'un sous-système type

Le fonctionnement de la sécurité d'un sous-système type, selon d'autres normes, doit être conforme au Tableau 4.

Tableau 4 – SIL exigé et *PFH* du sous-système type

IEC 62061 (IEC 61508)	IEC 62061	IEC 61508 ^a	ISO 13849 ^b	IEC 61496
<i>PFH</i>	SIL	au moins ...	au moins ...	au moins ...
$< 10^{-5}$	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 Une relation entre l'IEC 62061 et l'IEC 61511 (toutes les parties) ou l'ISO 26262 ne peut pas être prise pour hypothèse dans ce tableau.				
^a Cette colonne inclut les normes SIL qui satisfont aux contraintes architecturales de l'IEC 61508, comme l'IEC 61800-5-2 et l'IEC 60947-5-3.				
^b Ne s'applique pas aux sous-systèmes utilisant des composants complexes, sauf s'ils satisfont aux exigences de l'IEC 61508 ou des normes de produits applicables en matière de sécurité fonctionnelle. Le niveau de performance b ne correspond pas à SIL 1 dans le cas d'une structure de catégorie B (ISO 13849-1).				

6.4 Détermination de l'intégrité de sécurité du SCS

6.4.1 Généralités

Le ou les SIL qui peuvent être atteints par le SCS doivent être pris en considération séparément pour chaque fonction de sécurité et doivent être déterminés à partir du niveau de SIL et de la *PFH* de chaque sous-système, comme suit:

- le SIL qui a été atteint est inférieur ou égal au SIL le plus bas de l'un quelconque des sous-systèmes, et
- le niveau de SIL est limité par la somme des valeurs de *PFH* de tous les sous-systèmes conformément au Tableau 3.

La Figure 7 présente un exemple de SCS avec une intégrité de sécurité SIL 2, bien que la valeur *PFH* globale soit adaptée à un niveau de SIL plus élevé.

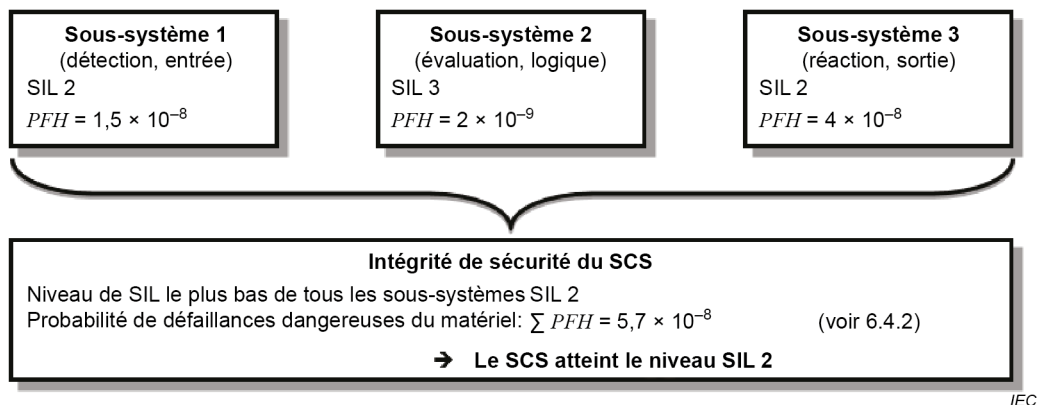


Figure 7 – Exemple d'intégrité de sécurité d'une fonction de sécurité reposant sur des sous-systèmes attribués en tant que SCS unique

NOTE Un SCS peut être une combinaison de sous-systèmes reposant sur différentes architectures.

6.4.2 PFH

La *PFH* de chaque fonction de sécurité due aux défaillances dangereuses aléatoires du matériel doit être inférieure ou égale à la *PFH* indiquée dans le Tableau 3 en ce qui concerne le niveau de SIL exigé, comme cela est indiqué dans la spécification des exigences de sécurité.

L'estimation de la *PFH* doit reposer sur la *PFH* de chaque sous-système correspondant, le cas échéant, pour les processus de communication de données numériques entre les sous-systèmes. La *PFH* du SCS est la somme des probabilités de défaillance dangereuse aléatoire du matériel de tous les sous-systèmes impliqués dans le fonctionnement de la fonction de sécurité et doit comprendre, le cas échéant, la probabilité maximale d'erreurs de transmission dangereuses (P_{TE}) dans le cas de la communication de données numériques:

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

NOTE 1 Cette approche repose sur la définition d'un sous-système qui établit que la défaillance d'un sous-système entraîne la défaillance du SCS (voir 6.3.1).

NOTE 2 Les aspects relatifs au câblage du matériel font partie de l'intégrité systématique et des défaillances possibles peuvent être détectées par des diagnostics.

NOTE 3 Pour la détermination du P_{TE} , voir par exemple l'IEC 61784-3.

6.5 Exigences pour l'intégrité de sécurité systématique du SCS

6.5.1 Exigences pour l'évitement des défaillances systématiques du matériel

Les mesures suivantes doivent s'appliquer, le cas échéant:

- le SCS doit être conçu et réalisé conformément au plan de sécurité fonctionnelle (voir 4.3);
- choix, combinaison, arrangements, assemblage et montage corrects des sous-systèmes, y compris les câbles, les conducteurs et toutes les interconnexions. L'interconnexion de câblage des sous-systèmes peut exiger des considérations relatives aux anomalies et des exclusions d'anomalies (voir 7.3.3);
- utilisation des SCS dans le cadre de la spécification du fabricant;
- utilisation de sous-systèmes aux caractéristiques de fonctionnement compatibles;

NOTE Voir également l'ISO 13849-2:2012, Annexes A, B, C et D.

- le SCS doit être installé et protégé selon l'IEC 60204-1, y compris la détection des défauts à la terre;

- f) les modes de fonctionnement non documentés de composants ne doivent pas être utilisés (par exemple les registres "réservés" d'un matériel programmable);
- g) prise en considération du mauvais usage prévisible, des changements ou modification(s) de l'environnement;
- h) les instructions du fabricant (y compris les exemples d'application, par exemple) des deux systèmes interconnectés (sorties du sous-système précédent et entrées du sous-système suivant) doivent être appliquées. Elles peuvent inclure:
 - les aspects liés au matériel (informations relatives à l'interface, blindage, niveau du signal, seuil de pression, impulsions d'essai, contraintes architecturales, par exemple),
 - les aspects liés aux logiciels (définition des télégrammes de communication de données, par exemple) et
 - les aspects liés à la couverture du diagnostic.

De plus, au moins une des techniques et/ou mesures suivantes doit s'appliquer prenant en compte la complexité du SCS et le ou les SIL des fonctions à réaliser par le SCS:

- i) la revue de conception du matériel du SCS (par exemple, par inspection ou lecture croisée): faire apparaître, par des examens et/ou des analyses, toutes les divergences entre la spécification et la mise en œuvre;

NOTE 1 Afin de faire apparaître les divergences entre la spécification et la mise en œuvre, tous les points de doute ou éventuels points faibles concernant la réalisation, la mise en œuvre et l'utilisation du produit sont documentés de façon à ce qu'ils puissent être résolus, prenant en compte que lors d'une procédure d'examen, l'auteur est passif et l'inspecteur est actif tandis que lors d'une procédure de lecture croisée, l'auteur est actif et l'inspecteur est passif.

- j) des outils d'aide tels que des ensembles de conception assistée par ordinateur capables de simulations ou d'analyses, et/ou l'utilisation d'outils de conception assistée par ordinateur afin d'exécuter des procédures de conception systématiques utilisant des éléments types déjà disponibles et soumis à l'essai.

NOTE 2 L'intégrité de ces outils peut être démontrée par des essais spécifiques, par un rapport détaillé sur une utilisation satisfaisante ou par une vérification indépendante de leurs résultats pour le SCS particulier en cours de conception.

- k) simulation: effectuer une assimilation complète et systématique de la conception d'un SCS en ce qui concerne les caractéristiques fonctionnelles, le dimensionnement et l'interaction corrects de ses sous-systèmes.

EXEMPLE Les fonctions du SCS peuvent être simulées sur ordinateur par un modèle comportemental de logiciel dans lequel les sous-systèmes particuliers ou éléments de sous-systèmes ont chacun leur propre comportement simulé, et la réponse du circuit dans lequel ils sont connectés est étudiée en observant les spécifications aux limites de chaque sous-système ou élément de sous-système.

6.5.2 Exigences pour la maîtrise des anomalies systématiques

Les mesures suivantes doivent s'appliquer:

- a) utilisation de l'absence de tension: le SCS doit être conçu de façon qu'en cas de perte de son alimentation, l'état de sécurité de la machine puisse être réalisé ou maintenu;
- b) mesures pour maîtriser l'effet des défaillances temporaires d'un sous-système: le SCS doit être conçu par exemple de façon que:
 - les effets d'une variation de l'alimentation (par exemple coupures, creux de tension) sur un sous-système particulier ou une partie d'un sous-système n'entraînent pas un phénomène dangereux (par exemple, une interruption de tension qui affecte un circuit de moteur ne doit pas provoquer une mise en marche intempestive lorsque l'alimentation est rétablie), et

NOTE 1 Voir aussi les exigences appropriées de l'IEC 60204-1. En particulier:

- la surtension ou la sous-tension peut être détectée suffisamment tôt de façon que toutes les sorties puissent être commutées en position de sécurité par le programme de mise hors tension ou basculées sur une seconde unité d'alimentation; et/ou

- si nécessaire, la surtension ou la sous-tension peut être détectée suffisamment tôt pour que l'état interne puisse être sauvegardé dans une mémoire non volatile, de façon que toutes les sorties puissent être commutées en position de sécurité par le programme de mise hors tension ou basculées sur une seconde unité d'alimentation.

Voir aussi les informations correspondantes de l'IEC 61131-2.

- les effets des interférences électromagnétiques de l'environnement physique ou d'un ou plusieurs sous-systèmes n'entraînent pas un phénomène dangereux;
- c) mesures pour maîtriser les effets des erreurs et autres effets liés à une quelconque communication de données, y compris les erreurs de transmission (par exemple, les répétitions, les suppressions, les insertions, les modifications du séquençement, l'altération, le retard et le masquage);

NOTE 2 Des informations complémentaires peuvent être obtenues dans le Tableau 1 de l'IEC 61784-3:2016 et en 7.4.11.2 de l'IEC 61508-2:2010.

NOTE 3 Le terme "masquage" signifie que le contenu exact d'un message n'est pas correctement identifié. Par exemple, un message provenant d'un composant qui n'est pas de sécurité est identifié incorrectement comme un message provenant d'un composant de sécurité.

- d) lorsqu'une défaillance dangereuse se produit sur une interface, la fonction réaction à l'anomalie doit être exécutée avant que le phénomène dangereux dû à cette anomalie ne puisse se produire. Lorsqu'une anomalie réduisant la tolérance aux anomalies du matériel à zéro se produit, cette réaction à l'anomalie doit être exécutée avant que la MTTR estimée ne soit dépassée (voir 3.2.39).

Les exigences du point d) s'appliquent aux interfaces qui sont les entrées et les sorties des sous-systèmes et toutes les autres parties de sous-systèmes qui comprennent du câblage ou exigent des opérations de câblage en phase d'intégration (par exemple, les appareils de coupure des signaux de sortie d'une barrière immatérielle, d'une sortie d'un capteur de position du protecteur).

NOTE 4 Il n'est pas exigé qu'un sous-système ou élément de sous-système détecte de lui-même une anomalie au niveau de sa ou ses sorties. La fonction réaction à l'anomalie peut aussi être initiée par tout sous-système subséquent à l'issue de l'exécution d'un essai de diagnostic.

6.6 Immunité électromagnétique

Aucune influence extérieure ne doit avoir d'impact sur la fonction des systèmes électriques ou électroniques relatifs à la sécurité d'une manière susceptible de donner lieu à un risque inacceptable. Des performances acceptables en ce qui concerne les perturbations électromagnétiques sont donc obligatoires. Une analyse de sécurité exhaustive doit inclure les effets des perturbations électromagnétiques et limites d'immunité électromagnétique exigés pour atteindre la sécurité fonctionnelle. Il convient de déduire ces limites en tenant compte de l'environnement électromagnétique et des niveaux d'intégrité de sécurité exigés.

Le SCS doit satisfaire aux exigences applicables de l'IEC 61000-1-2.

NOTE 1 Les niveaux d'immunité appropriés dans le cas des environnements industriels sont donnés dans l'IEC 61326-3-1 ou l'IEC 61000-6-7 au moins.

NOTE 2 Si un sous-système a été conçu selon une norme appropriée de produit relative à la sécurité (l'IEC 61496-1, par exemple), l'IEC 61326-3-1 ou l'IEC 61000-6-7, les informations peuvent être fournies avec le sous-système, ce qui facilite la vérification des exigences de niveau du SCS par une analyse.

NOTE 3 Des principes de conception sont disponibles dans les normes CEM, mais les normes de sécurité fonctionnelle exigent des niveaux d'immunité plus élevés. Il est important de reconnaître que des niveaux d'immunité plus élevés que ceux spécifiés (ou des exigences d'immunité supplémentaires à celles spécifiées) dans ces normes peuvent s'avérer nécessaires pour des emplacements particuliers ou lorsque le matériel est destiné à être utilisé dans des environnements électromagnétiques plus sévères ou différents.

6.7 Paramétrisation manuelle liée au logiciel

6.7.1 Généralités

Certains sous-systèmes relatifs à la sécurité ou SCS ont besoin d'être paramétrés pour réaliser une fonction ou sous-fonction de sécurité. Par exemple, un convertisseur avec des sous-fonctions intégrées doit être paramétré avec un outil de configuration PC, en fonction de la limite de vitesse de sécurité supérieure. De la même manière, pour établir correctement la zone de détection d'un lecteur de codes à barres laser, des paramètres tels que l'angle et la distance peuvent avoir besoin d'être configurés selon la documentation de sécurité du fabricant et l'appréciation du risque de la machine.

Les exigences en matière de paramétrisation manuelle liée au logiciel ont pour objet de garantir que les paramètres relatifs à la sécurité, spécifiés pour une fonction de sécurité ou une sous-fonction, sont correctement transférés dans le matériel qui exécute la fonction de sécurité ou une sous-fonction. Différentes méthodes peuvent être appliquées pour définir des paramètres. Une paramétrisation fondée sur un commutateur DIP peut même être utilisée pour définir ou modifier des paramètres relatifs à la sécurité. Toutefois, les outils PC avec des logiciels de paramétrisation dédiés, souvent appelés outils de configuration ou de paramétrisation, prennent de plus en plus d'importance. Le présent paragraphe se limite à la paramétrisation logicielle manuelle réalisée et maîtrisée par une personne autorisée.

NOTE 1 La paramétrisation relative à la sécurité réalisée automatiquement sans interaction humaine (en s'appuyant sur des signaux d'entrée, par exemple) n'est pas prise en considération dans le présent 6.7.

NOTE 2 La commande directe d'une machine par un opérateur (le contrôle de vitesse d'un chariot élévateur, par exemple) n'est pas considérée comme une paramétrisation manuelle telle que décrite dans le présent paragraphe.

NOTE 3 S'il s'agit d'un outil de configuration ou de paramétrisation type selon l'IEC 61508-3 (avec son sous-système dédié, par exemple), aucune défaillance dangereuse due aux influences indiquées en 6.7.2 ou à d'autres influences n'est, par hypothèse, raisonnablement prévisible. Les exigences de 6.7.5 s'appliquent lorsqu'une paramétrisation manuelle liée au logiciel est réalisée avec l'outil type.

6.7.2 Influences sur les paramètres relatifs à la sécurité

Lors d'une paramétrisation manuelle liée au logiciel, les paramètres peuvent être affectés par plusieurs influences, telles que:

- les erreurs d'entrée de données par la personne chargée de la paramétrisation;
- les anomalies du logiciel de l'outil de paramétrisation;
- les anomalies d'autres logiciels et/ou services fournis avec l'outil de paramétrisation;
- les anomalies du matériel de l'outil de paramétrisation;
- les anomalies lors de la transmission des paramètres entre l'outil de paramétrisation et le SCS ou un sous-système;
- les anomalies du SCS ou d'un sous-système pour stocker correctement les paramètres transmis;
- les interférences systématiques lors du processus de paramétrisation (interférences électromagnétiques ou perte de puissance, par exemple);
- les interférences dues à des influences ou des facteurs extérieurs, comme des interférences électromagnétiques ou la perte de puissance (aléatoire).

Si aucune mesure n'est appliquée pour remédier, éviter ou maîtriser les défaillances dangereuses potentielles provoquées par les influences indiquées ci-dessus, cela peut donner lieu à ce qui suit:

- les paramètres ne sont pas mis à jour par le processus de paramétrisation, en totalité ou en partie, sans en informer la personne chargée de la paramétrisation;
- les paramètres sont incorrects, en totalité ou en partie;
- les paramètres sont appliqués à un dispositif incorrect, comme lorsque la transmission des paramètres est réalisée par l'intermédiaire d'un réseau filaire ou sans fil.

6.7.3 Exigences relatives à la paramétrisation manuelle liée au logiciel

La paramétrisation manuelle liée au logiciel doit utiliser un outil dédié fourni par le fabricant ou le fournisseur du SCS ou du ou des sous-systèmes associés. Cet outil doit posséder sa propre identification (nom, version, etc.). Le SCS ou le ou les sous-systèmes associés et l'outil de paramétrisation doivent avoir la capacité d'empêcher toute modification non autorisée (à l'aide d'un mot de passe dédié, par exemple).

La paramétrisation alors que la machine est en fonctionnement ne doit être admise que si elle ne provoque pas de situation dangereuse.

L'utilisation d'un SCS ou d'un sous-système type permettant de procéder à la paramétrisation manuelle liée au logiciel a pour objet d'empêcher la défaillance dangereuse due aux influences indiquées en 6.7.2 ou à d'autres influences raisonnablement prévisibles.

Il est possible de satisfaire aux exigences en utilisant un SCS ou sous-système type, ou la conception du SCS ou du sous-système utilisé doit être conforme au présent document. Les aspects de la paramétrisation doivent être pris en compte dans la validation du SCS.

Les exigences suivantes doivent être satisfaites.

- a) La conception de la paramétrisation manuelle liée au logiciel doit être considérée comme un aspect relatif à la sécurité de la conception du SCS décrite dans une spécification des exigences de sécurité, par exemple la spécification des exigences de sécurité du logiciel (voir 8.3.2.2 et 8.4.2.2).
- b) Le SCS ou le sous-système doit fournir des moyens de vérifier le caractère plausible des données (vérifications des limites de données, du format et/ou des valeurs d'entrée logiques, par exemple).
- c) L'intégrité de toutes les données utilisées pour la paramétrisation doit être maintenue. Cela doit être réalisé en mettant en œuvre des mesures pour:
 - maîtriser la plage des entrées valides;
 - maîtriser l'altération des données avant transmission;
 - maîtriser les effets des erreurs du processus de transmission des paramètres;
 - maîtriser les effets d'une transmission de paramètres incomplète;
 - maîtriser les effets des anomalies et des défaillances des matériel et logiciel de la paramétrisation, et
 - maîtriser les effets de l'interruption de l'alimentation.
- d) L'outil de paramétrisation doit satisfaire à l'ensemble des exigences pertinentes pour un sous-système selon l'IEC 61508 afin d'assurer une paramétrisation correcte.
- e) Outre d), une procédure particulière doit être utilisée pour le réglage des paramètres relatifs à la sécurité. Cette procédure doit comprendre la confirmation des paramètres d'entrée du SCS par soit:
 - la retransmission des paramètres modifiés vers l'outil de paramétrisation, soit
 - d'autres moyens de confirmation de l'intégrité des paramètres,

ainsi que la confirmation ultérieure (par exemple, par une personne qualifiée convenablement et au moyen d'une vérification automatique par un outil de paramétrisation). Les nouvelles valeurs des paramètres relatifs à la sécurité ne doivent pas être activées avant de reconnaître et de confirmer les modifications.

NOTE L'activité de confirmation revêt une importance particulière lorsqu'un outil logiciel de paramétrisation utilise un dispositif qui n'est pas particulièrement destiné à cet usage (par exemple, ordinateur personnel ou appareil équivalent).

Les modules de logiciel utilisés pour le codage/décodage dans les processus de transmission/retransmission, ainsi que les modules de logiciel utilisés pour la visualisation

des paramètres relatifs à la sécurité destinée à l'utilisateur, doivent au minimum utiliser la diversité dans la ou les fonctions destinées à éviter les défaillances systématiques.

6.7.4 Vérification de l'outil de paramétrisation

Les vérifications suivantes doivent au moins être réalisées pour vérifier la fonctionnalité de base de l'outil de paramétrisation:

- vérification du réglage correct pour chaque paramètre relatif à la sécurité (valeurs minimale, maximale et valeurs représentatives);
- vérification que les paramètres relatifs à la sécurité sont soumis à un contrôle de vraisemblance, par exemple par la détection de valeurs invalides, etc.;
- vérification que des moyens sont prévus pour éviter la modification non autorisée des paramètres relatifs à la sécurité.

NOTE L'activité de vérification revêt une importance particulière lorsque la paramétrisation est réalisée en utilisant un dispositif qui n'est pas particulièrement destiné à cet usage (par exemple ordinateur personnel ou appareil équivalent).

6.7.5 Performances de la paramétrisation manuelle liée au logiciel

La paramétrisation manuelle liée au logiciel doit être réalisée à l'aide de l'outil de paramétrisation dédié fourni par le fabricant ou le fournisseur du SCS ou du ou des sous-systèmes associés, et doit être documentée conformément aux exigences indiquées dans les informations d'utilisation. Ces informations peuvent provenir de différentes parties. Voir également 10.3 (informations d'utilisation). Des mesures de prévention contre les accès non autorisés doivent être activées et utilisées.

La paramétrisation initiale et les modifications subséquentes qui lui sont apportées doivent être documentées. La documentation doit inclure:

- a) la date de la paramétrisation initiale ou de la modification;
- b) les données ou le numéro de version de l'ensemble de données;
- c) le nom de la personne qui a procédé à la paramétrisation;
- d) une indication de l'origine des données utilisées (jeux de paramètres prédéfinis, par exemple);
- e) une identification claire des paramètres relatifs à la sécurité;
- f) une identification claire des SCS soumis à des réglages spécifiques de paramétrisation.

6.8 Aspects liés à la sécurité

La sécurité couvre les attaques volontaires du matériel, des programmes d'application et des logiciels associés, ainsi que les événements imprévus résultant d'une erreur humaine.

NOTE 1 Les aspects liés à la sécurité sont pris en considération dans le cycle de vie de sécurité de la machine (ou un niveau système plus élevé) et tout au long du cycle de vie de la machine.

NOTE 2 Le présent document ne fournit pas d'exigences particulières sur les aspects liés à la sécurité, mais des recommandations relatives à ces aspects sont fournies dans les normes IEC TR 63074, ISA TR84.00.09, ISO/IEC 27001:2013, ISO TR 22100-4 et IEC 62443 (toutes les parties).

Lorsque des mesures préventives de sécurité sont appliquées, elles ne doivent pas avoir compromettre l'intégrité de sécurité (augmentation du temps de réponse, par exemple). Cela peut exiger une analyse en équipe multidisciplinaire itérative.

Lorsque les mesures préventives de sécurité mises en œuvre dans le SCS sont déclarées, des informations doivent être fournies, selon le cas.

6.9 Aspects des essais périodiques

Les essais périodiques de la fonction de sécurité ou des sous-fonctions ont deux principaux objectifs:

- les essais périodiques confirment, à un instant donné, que la fonction soumise à l'essai ne présente aucune anomalie;
- les essais périodiques associés à des examens permettent de vérifier que les conditions aux limites des chiffres de fiabilité du matériel sont respectées.

En règle générale, il existe deux types d'essais périodiques:

- les essais de diagnostic sont réalisés automatiquement (initiés automatiquement ou manuellement) et souvent (en fonction du temps de sécurité du processus et du taux de sollicitation);

NOTE 1 Les essais périodiques peuvent s'appliquer à une sous-fonction ou à une fonction de sécurité.

- les essais périodiques tentent de vérifier l'ensemble de la fonction, généralement en simulant une condition dangereuse pour les capteurs ou au moins les unités logiques. De même, des examens de vieillissement et de dégradation des composants sont réalisés dans le cadre des essais périodiques.

NOTE 2 Les défaillances dangereuses qui ne peuvent pas être détectées par les diagnostics sont considérées comme étant des défaillances dangereuses non détectées (taux de défaillance associé λ_{DU}). Elles ne peuvent être détectées que par l'essai périodique.

Pour utiliser des essais périodiques comme assurance d'intégrité de sécurité, les conditions suivantes doivent être satisfaites:

- dans la procédure d'essai, une réaction à l'anomalie doit être mise en œuvre pour placer les parties concernées de la machine dans un état de sécurité par suite de la détection d'une anomalie;

NOTE 3 La nature d'une réaction à l'anomalie peut être différente pour le diagnostic et l'essai périodique. Cela dépend également du mode de sollicitation et de l'architecture. Pour l'architecture des fonctions avec une tolérance aux anomalies du matériel de zéro (HFT 0) et un mode à forte sollicitation ou continu, il est en général exigé d'arrêter immédiatement les machines.

- l'intervalle d'essai doit être adapté pour révéler les défaillances en fonction du taux de sollicitation;
- pour les essais de diagnostic, voir également 7.4.3 pour des exigences particulières.

7 Conception et développement d'un sous-système

7.1 Généralités

Le sous-système doit être conçu selon sa spécification des exigences de sécurité (voir 5.2), y compris essentiellement:

- les exigences fonctionnelles;
- les exigences pour l'intégrité de sécurité du matériel:
 - les contraintes architecturales (voir 7.4) et
 - *PFH* (voir 7.6);
- les exigences d'intégrité systématique (voir 7.3.2 et une estimation de la défaillance de cause commune (CCF) à l'Annex E);
- les exigences relatives au comportement du sous-système sur détection d'anomalie (réaction à l'anomalie) (voir 7.4.3);
- les exigences en matière de logiciel (voir l'Article 8).

Les informations suivantes du Tableau 5 doivent être disponibles, le cas échéant, pour chaque sous-système lors de la conception et du développement.

Tableau 5 – Informations pertinentes pour chaque sous-système

Description fonctionnelle	
1)	Une description fonctionnelle de la ou des fonctions et interfaces du sous-système
Informations matérielles	
2)	Les taux de défaillance estimés (en raison de défaillances aléatoires du matériel et des modes de défaillance) pour chaque élément de sous-système qui peuvent provoquer une défaillance dangereuse du sous-système (voir l'Annexe C)
3)	Tous les essais et/ou toutes les exigences de maintenance
4)	La probabilité d'erreurs de communication dangereuses dans le cas de processus de communication de données numériques, le cas échéant
Conditions environnementales	
5)	L'environnement et les conditions de fonctionnement qu'il convient d'observer afin de maintenir la validité des taux de défaillance estimés dus à des défaillances aléatoires du matériel
6)	La durée de fonctionnement utile (voir 7.3.4.2) du sous-système qu'il convient de ne pas dépasser, afin de maintenir la validité des taux de défaillance estimés dus à des défaillances aléatoires du matériel
Informations relatives à la conception	
7)	La couverture du diagnostic et/ou la proportion de défaillances en sécurité et l'intervalle entre essais de diagnostic (voir 7.4.3 et 7.4.4)
8)	Les limitations concernant l'application du sous-système qu'il convient d'observer afin d'éviter ou de maîtriser les défaillances systématiques
9)	Les informations exigées pour identifier la configuration matérielle et logicielle du sous-système
10)	Le SIL le plus élevé qui peut être revendiqué pour une fonction de sécurité à l'étude qui utilise le sous-système sur la base: – des contraintes architecturales, – des mesures et techniques utilisées pour éviter ou maîtriser les anomalies systématiques introduites lors de la conception et de la réalisation du matériel et du logiciel du sous-système, et – des caractéristiques de conception qui rendent le sous-système tolérant aux anomalies systématiques. NOTE Un sous-système peut mettre en œuvre des sous-fonctions de plusieurs fonctions de sécurité avec un niveau SIL différent.

7.2 Conception de l'architecture d'un sous-système

L'architecture d'un sous-système est définie par un processus de décomposition fonctionnelle qui s'apparente à celui de la fonction de sécurité complète donnant lieu à l'architecture SCS (voir 6.3.2). La sous-fonction spécifique du sous-système peut être décomposée en sous-fonctions de l'ordre inférieur suivant qui sont assignées aux éléments de sous-système.

Il en résulte qu'un ensemble d'éléments de sous-système peut être défini, lequel satisfait aux exigences fonctionnelles et aux exigences d'intégrité de la sous-fonction.

NOTE 1 Un sous-système peut être conçu à l'aide d'un seul élément de sous-système.

NOTE 2 La décomposition en élément(s) de sous-système peut être un processus itératif.

NOTE 3 La défaillance d'un élément de sous-système ne donne pas nécessairement lieu à une défaillance du sous-système ou de la sous-fonction. Lorsque les éléments de sous-système font partie intégrante des canaux redondants, la défaillance d'un seul élément ne donne pas lieu à celle de la fonction de sécurité.

La conception de l'architecture de sous-système doit être documentée en ce qui concerne ses éléments de sous-système et leurs relations (schéma de circuit avec description, schéma fonctionnel relatif à la sécurité, par exemple).

Le ou les sous-systèmes incorporant des composants complexes doivent se conformer aux normes de produits appropriées ou à l'IEC 61508-2 et à l'IEC 61508-3 pour autant qu'elles sont applicables pour le niveau SIL exigé, et la conception doit utiliser le parcours 1_H (voir l'IEC 61508-2:2010, 7.4.4.2) pour le mode à forte sollicitation et/ou continu. Lorsque la conception d'un sous-système comporte ce type de composant complexe en tant qu'élément de sous-système, ce dernier peut être considéré comme un composant de faible complexité dans le contexte d'une conception de sous-système, étant donné que les modes de défaillance pertinents, le comportement sur détection d'une anomalie, le taux de défaillance et d'autres informations relatives à la sécurité sont connus. De tels composants doivent être utilisés uniquement selon leur spécification et les informations appropriées relatives à leur utilisation, données par leur fabricant.

NOTE 4 Le présent document considère que la conception des sous-systèmes ou éléments de sous-système électroniques programmables complexes est conforme aux exigences appropriées de l'IEC 61508 et utilise le parcours 1_H (voir l'IEC 61508-2:2010, 7.4.4.2).

7.3 Exigences pour le choix et la conception du sous-système et des éléments de sous-système

7.3.1 Généralités

Il existe deux types d'exigences relatives aux sous-systèmes et éléments de sous-système:

- les exigences qualitatives: intégrité systématique; prise(s) en considération des anomalies et exclusion(s) des anomalies;
- les exigences quantitatives: taux de défaillance et autres paramètres pertinents.

Les exigences qualitatives sont définies dans les Paragraphes 7.3.2 et 7.3.3 suivants. Si elles ne sont pas établies de manière explicite, ces exigences s'appliquent indépendamment de l'exigence SIL à la fonction de sécurité du SIL 1 au SIL 3.

NOTE Le SIL 4 n'est pas pris en considération dans le présent document, car il n'est pas approprié aux exigences de réduction du risque associées aux machines. Pour les exigences applicables au SIL 4, voir l'IEC 61508-1 et l'IEC 61508-2.

Les exigences quantitatives sont décrites en 7.4 en termes généraux, et pour la détermination de la *PFH*, voir 6.3.2 et 7.6.

7.3.2 Intégrité systématique

7.3.2.1 Généralités

Les exigences d'intégrité de sécurité systématique d'un sous-système sont respectées en satisfaisant aux exigences de 7.3.2.2 et 7.3.2.3 et sont identiques pour SIL 1, SIL 2 et SIL 3.

NOTE Le sous-système peut être partitionné en éléments de sous-système, préconçus selon l'IEC 61508, avec différents niveaux de capacité systématique. Ensuite, la capacité systématique d'un élément de sous-système peut potentiellement limiter le SIL de son sous-système. Pour des détails supplémentaires, voir l'IEC 61508-2.

7.3.2.2 Exigences pour l'évitement des défaillances systématiques

Les mesures suivantes doivent toutes s'appliquer, le cas échéant:

- choix, combinaison, arrangements, dispositions, assemblage et installation appropriés des composants, y compris les câbles, les conducteurs et toutes les interconnexions: utilisation des notes d'application du fabricant, par exemple le manuel d'utilisation, les instructions d'installation, les spécifications et l'utilisation des règles de l'art (l'IEC 60204-1, par exemple);
- utilisation du sous-système et des éléments de sous-système dans le cadre de la spécification du fabricant et selon les instructions d'installation;
- compatibilité: utilisation de composants ayant des caractéristiques de fonctionnement compatibles;

- résistance à des conditions environnementales spécifiées:
concevoir le sous-système de façon qu'il soit capable de travailler dans tous les environnements prévus et dans toutes les conditions défavorables prévisibles (dans les limites d'utilisation définies), par exemple, température, humidité, vibrations et interférences électromagnétiques;
- utilisation de composants selon une norme applicable et dont les modes de défaillance sont bien définis: pour réduire le risque d'anomalies non détectées par l'utilisation de composants ayant des caractéristiques spécifiées;

NOTE 1 Les composants tels que les soupapes hydrauliques ou pneumatiques peuvent exiger une commutation cyclique pour éviter le mode de défaillance de non-commutation ou l'augmentation inacceptable de temps de commutation. Dans ce cas, un essai périodique peut s'avérer nécessaire.

- utilisation de matériaux appropriés et de fabrication adéquate:
choix des matériaux, méthodes et procédés de fabrication en relation avec par exemple les contraintes, la durabilité, l'élasticité, la friction, l'usure, la corrosion, la température, la conductivité, la rigidité diélectrique;
- dimensions et formes correctes:
considérer les effets, par exemple, des contraintes, de la traction, de la fatigue, de la température, de la rugosité de surface, des tolérances de fabrication.

NOTE 2 IEC 61508-2:2010, Annexe F spécifie les techniques et mesures permettant d'éviter les défaillances systématiques pendant la conception et le développement des circuits intégrés spécifiques à l'application (ASIC), des réseaux de portes programmables sur site (FPGA), des réseaux logiques programmables (PLD), etc.

NOTE 3 Les Tableaux B.1 à B.5 de l'IEC 61508-2:2010, Annexe B, donnent les techniques et mesures permettant d'éviter les défaillances des systèmes relatifs à la sécurité, qui peuvent être utiles lors des phases de spécification, de conception, d'intégration, de fonctionnement, d'entretien et de validation.

NOTE 4 Les Annexes A à D de l'ISO 13849-2:2012 fournissent des principes aux systèmes mécaniques, pneumatiques, hydrauliques et électriques.

De plus, au moins une des mesures suivantes doit être appliquée, le cas échéant:

- a) revue de conception du matériel (par examen ou lecture croisée, par exemple):
faire apparaître, par des examens et/ou des analyses, les divergences entre la spécification et la mise en œuvre;

NOTE 5 Afin de faire apparaître les divergences entre la spécification et la mise en œuvre, tous les points de doute ou éventuels points faibles de la réalisation, la mise en œuvre et l'utilisation du produit sont documentés de manière à pouvoir les résoudre. Lors d'une procédure d'examen, l'auteur est passif et l'inspecteur est actif, tandis que lors d'une procédure de lecture croisée, l'auteur est actif et l'inspecteur est passif.

- b) outils de conception assistée par ordinateur capables de simulation ou d'analyse:
effectuer la procédure systématique de conception et inclure les éléments de construction automatiques appropriés déjà disponibles et soumis à l'essai;

NOTE 6 Ces outils peuvent être qualifiés par des essais spécifiques ou par un rapport détaillé sur une utilisation satisfaisante ou par une vérification indépendante de leurs résultats pour le sous-système particulier en cours de conception.

- c) simulation:
effectuer une simulation systématique de la conception d'un sous-système en ce qui concerne les caractéristiques fonctionnelles et le dimensionnement correct de ses composants.

NOTE 7 La fonction du sous-système peut être simulée sur ordinateur par un modèle comportemental de logiciel dans lequel les composants particuliers du circuit ont chacun leur propre comportement simulé, et la réponse du sous-système dans lequel ils sont connectés est étudiée en observant les spécifications aux limites de chaque composant.

7.3.2.3 Exigences pour la maîtrise des anomalies systématiques

Les mesures suivantes doivent toutes s'appliquer, le cas échéant:

- a) mesures pour maîtriser les effets d'un claquage de l'isolant, de variations et d'interruptions de tension, d'une surtension, d'une sous-tension: le comportement du sous-système en réponse à un claquage de l'isolant, à des variations et interruptions de tension, à des

conditions de surtension et de sous-tension doit être prédéterminé de façon que le sous-système puisse réaliser ou maintenir un état de sécurité;

NOTE 1 De plus amples informations peuvent être obtenues dans l'IEC 60204-1 et l'IEC 61508-7:2010, Article A.8.

- b) mesures pour maîtriser ou éviter les effets de l'environnement physique (par exemple, la température, l'humidité, l'eau, les vibrations, la poussière, les substances corrosives, les interférences électromagnétiques et leurs effets): le comportement du sous-système en réponse aux effets de l'environnement physique doit être prédéterminé de façon que le SCS puisse réaliser ou maintenir un état de sécurité. Voir également, par exemple, l'IEC 60529, l'IEC 60204-1 et l'IEC 60721 (toutes les parties);
- c) mesures pour maîtriser ou éviter les effets d'un accroissement ou d'une diminution de la température si des variations de température sont susceptibles de se produire: il convient que le sous-système soit conçu de façon que, par exemple, une surchauffe puisse être détectée avant que le sous-système ne commence à fonctionner en dehors de sa spécification;

NOTE 2 De plus amples informations peuvent être obtenues dans l'IEC 61508-7:2010, Article A.10.

- d) mesures pour maîtriser les effets d'une rupture de tuyau, de variations et d'interruptions de pression, d'une pression trop basse ou trop élevée: le comportement du sous-système en réponse à une rupture de tuyau, à des variations et interruptions de pression, à une pression trop basse ou trop élevée doit être prédéterminé de façon que le sous-système puisse réaliser ou maintenir un état de sécurité.

NOTE 3 D'autres informations peuvent être obtenues dans l'ISO 4414:2010 pour les systèmes pneumatiques ou l'ISO 4413 pour les systèmes hydrauliques.

Lorsqu'une alimentation TBTP/TBTS (voir l'IEC 60364-4-41) est utilisée, la surtension à la sortie en cas de premier défaut doit être prise en compte dans l'analyse des effets de la surtension, y compris la possibilité de défaillance de cause commune.

NOTE 4 Des plages de surtensions sont données en exemple dans l'IEC 60950-1, l'IEC 61204-7, l'IEC 62477 (toutes les parties) et l'IEC 60449.

En complément, les principes de sécurité de base suivants, selon le cas, doivent s'appliquer pour la maîtrise des défaillances systématiques:

- utilisation de l'absence de tension:
il convient de concevoir le sous-système de façon qu'en cas de perte de son alimentation, l'état de sécurité puisse être réalisé ou maintenu;

NOTE 5 Pour plus d'informations, voir l'ISO 13849-2.

- mesures pour maîtriser les effets des erreurs et autres effets liés à un quelconque processus de communication de données (voir l'IEC 61508-2:2010, 7.4.11).

Selon l'architecture choisie du sous-système, les principes de sécurité éprouvés suivants, selon le cas, doivent s'appliquer à l'élément de sous-système pour la maîtrise des défaillances systématiques:

- détection des défaillances par des essais automatiques;
- essais comparatifs du matériel redondant;

NOTE 6 Pour plus d'informations, voir l'ISO 12100:2010, 6.2.12.4.

- diversité du matériel;
- fonctionnement en mode positif (par exemple, un interrupteur de fin de course est actionné si un protecteur est ouvert);
- contacts mécaniquement liés;
- action d'ouverture directe;
- mode de défaillance orienté;

NOTE 7 Pour plus d'informations, voir l'ISO 12100:2010, 6.2.12.3.

- le surdimensionnement selon un facteur approprié peut améliorer la fiabilité et un facteur approprié de surdimensionnement doit être déterminé.

NOTE 8 Pour plus d'informations, voir l'ISO 13849-2 et l'Annexe A de l'IEC 61508-2:2010.

7.3.2.4 Immunité électromagnétique

La conception du sous-système doit satisfaire aux exigences de 6.6.

7.3.2.5 Aspects liés à la sécurité

La conception du sous-système doit satisfaire aux exigences de 6.8.

7.3.3 Prise en considération et exclusion des anomalies

7.3.3.1 Généralités

Tous les éléments de sous-système doivent être conçus pour obtenir la spécification des exigences de sécurité exigée. L'aptitude à résister aux anomalies doit être appréciée. Sauf spécification contraire explicite, les exigences du présent Article 7 s'appliquent indépendamment de l'intégrité de sécurité exigée de la fonction de sécurité.

7.3.3.2 Prise en considération des anomalies

Pour estimer l'aptitude des éléments de sous-système à atteindre un certain état de sécurité, chacun d'eux doit être analysé pour déterminer toutes les anomalies afférentes et leurs modes de défaillance correspondants. Qu'une défaillance soit une défaillance en sécurité ou une défaillance dangereuse dépend du SCS et des fonctions de sécurité prévues, y compris la fonction réaction à l'anomalie.

Une technique d'analyse comme l'analyse des modes de défaillance et de leurs effets (AMDE, voir l'IEC 60812) l'analyse par arbre de pannes (AAP, voir l'IEC 61025) ou l'analyse par arbre d'événements (AAE, voir l'IEC 62502) peut être utilisée pour déterminer les anomalies qui doivent être prises en considération pour ces composants.

La probabilité de chaque mode de défaillance doit être déterminée sur la base de la probabilité des anomalies associées en prenant en compte l'usage prévu, et peut être déduite de sources telles que:

- les données de taux de défaillance suffisamment fiables issues du retour d'expérience du fabricant et appropriées à l'usage prévu;
- les données de défaillance de composants provenant d'une origine industrielle reconnue et appropriées à l'usage prévu;
- les données de mode de défaillance;
- les données de taux de défaillance déduits des résultats des essais et des analyses.

En général, les critères d'anomalie suivants doivent être pris en compte:

- si, en conséquence d'une anomalie, d'autres composants tombent en panne, la première anomalie et toutes les anomalies suivantes doivent être considérées comme un premier défaut (appelé anomalie indépendante);
- au moins deux anomalies séparées ayant une cause commune doivent être considérées comme un premier défaut (appelé défaillance de cause commune);
- l'occurrence simultanée d'au moins deux anomalies ayant une cause distincte est considérée comme très peu probable, et il n'est donc pas nécessaire de la prendre en considération.

7.3.3.3 Exclusion d'anomalie

Il n'est pas toujours possible d'évaluer les sous-systèmes sans prendre pour hypothèse que certaines anomalies peuvent être exclues. L'exclusion d'anomalie est un compromis entre les exigences de sécurité technique et la possibilité théorique d'occurrence d'une anomalie.

L'exclusion d'anomalie peut reposer sur:

- l'improbabilité technique d'occurrence de certaines anomalies,
- l'expérience technique généralement acceptée, indépendante de l'application considérée, et
- les exigences techniques relatives à l'application et aux phénomènes dangereux spécifiques.

L'exclusion d'anomalie est uniquement applicable pour certaines défaillances d'un élément, et il revient au concepteur (fabricant ou intégrateur) de prouver l'exclusion des anomalies respectives en fonction des limites fixées par la conception et l'utilisation. Ce type d'exclusions d'anomalie est uniquement possible si l'improbabilité technique de leur occurrence peut être justifiée par des lois de science physique connues. De telles exclusions d'anomalies doivent être justifiées et documentées.

L'application de l'exclusion d'anomalie à certaines anomalies d'un élément se trouvant à l'intérieur d'un sous-système ne dispense pas d'appliquer des mesures systématiques.

Certaines anomalies peuvent être exclues par le fabricant, et d'autres par l'intégrateur de sous-système.

L'exclusion d'anomalie est un principe visant à limiter la défaillance d'un composant/sous-système; d'autres méthodes sont également possibles (architectures, limitation des défaillances systématiques, par exemple).

Le type d'anomalie exclue doit faire l'objet d'une caractérisation particulière. Il n'est pas acceptable d'établir simplement qu'un composant ne se casse pas, ne se déforme pas ou ne se dégrade pas en raison de l'usure. Il est nécessaire d'établir l'influence directe dans le cadre de laquelle le composant ne se casse pas, ne se déforme pas ou ne se dégrade pas en raison de l'usure. Par exemple, le composant ne présente aucune anomalie lorsqu'il est soumis à une force de X Newtons exercée dans la direction Y.

L'exclusion d'anomalie doit être justifiable dans tous les environnements industriels prévus, y compris en matière de température, de pression, de vibrations, de pollution, d'atmosphère corrosive, etc.

NOTE Des informations utiles relatives aux exclusions d'anomalie sont disponibles dans l'ISO 13849-2:2012, Annexes A à D.

L'exclusion d'anomalie peut uniquement être appliquée pour l'ensemble du sous-système si toutes les défaillances dangereuses d'un sous-système peuvent être exclues.

LIMITATION: Pour certaines applications, il n'est pas prévu de pouvoir exclure toutes les défaillances selon un niveau de confiance suffisant de SIL 3. La liste non exhaustive suivante donne une indication des sous-systèmes (non préconçus) avec une tolérance aux anomalies du matériel (HFT) de zéro et pour lesquels des exclusions d'anomalie ont été appliquées aux anomalies susceptibles d'entraîner une défaillance dangereuse dans laquelle un niveau maximal de SIL 2 peut être approprié, à condition de donner une justification suffisante:

- interrupteur de position avec aspects mécaniques avec HFT de 0;
- fuite d'une vanne hydraulique (si la fuite est une défaillance dangereuse).

NOTE Cette limitation ne s'applique pas aux sous-systèmes types utilisés à l'intérieur de leur spécification.

7.3.3.4 Essais fonctionnels pour détecter l'accumulation d'anomalies et les anomalies non détectées

Dans un système redondant, une accumulation d'anomalies dans le temps peut donner lieu à une perte de la fonction de sécurité. Dans un système à simple canal, les anomalies non détectées peuvent également donner lieu à une perte de la fonction de sécurité.

Pour un SCS à technologie non électronique et utilisant la surveillance automatique pour obtenir la couverture du diagnostic nécessaire au service des performances de sécurité exigées, la fonction de surveillance ne peut pas être possible, sauf en cas de changement d'état (à chaque cycle de fonctionnement, par exemple). Si, dans ce cas, le fonctionnement n'est que peu fréquent, la probabilité d'occurrence d'une anomalie non détectée augmente. Si un essai fonctionnel s'avère nécessaire pour détecter une éventuelle accumulation d'anomalies ou une anomalie non détectée avant la sollicitation suivante, il doit être réalisé dans les limites des intervalles d'essai suivants:

- au moins une fois par mois pour le niveau SIL 3;
- au moins une fois par an pour le niveau SIL 2.

EXEMPLE: Le système de commande d'une machine peut solliciter ces essais aux intervalles exigés (par affichage visuel ou un témoin lumineux, par exemple) et peut surveiller les essais et arrêter la machine si l'essai est ignoré ou n'est pas concluant.

7.3.4 Taux de défaillance de l'élément de sous-système

7.3.4.1 Généralités

La probabilité mathématique de défaillance d'un élément de sous-système peut être caractérisée par l'un des trois paramètres suivants: λ (Lambda), $MTTF$ (durée moyenne de fonctionnement avant défaillance) ou B_{10} .

NOTE Bien que les paramètres ci-dessus puissent être fournis en plusieurs formats valables, les formats classiques sont les suivants.

- λ : défaillances par heure;
- $MTTF$: durée moyenne de fonctionnement avant défaillance, exprimée en années;
- B_{10} : périodicités de démarrage des composants d'usure.

Pour estimer les paramètres d'un élément de sous-système, la procédure hiérarchique de détermination des données doit être, dans l'ordre indiqué:

- a) utiliser les données du fabricant;
- b) utiliser l'Annexe C du présent document;
- c) choisir un $MTTF_D$ de dix ans.

Les données peuvent être fournies sous la forme de valeurs par rapport aux défaillances dangereuses (λ_D , $MTTF_D$, B_{10D}) ou par rapport à toutes les défaillances (λ , $MTTF$, B_{10}).

Pour déterminer les défaillances dangereuses à partir des défaillances globales, il convient de tenir compte des différents modes de défaillance de l'élément de sous-système. En règle générale, par hypothèse tous les modes de défaillance ne donnent pas lieu à une défaillance dangereuse. Cela dépend essentiellement de l'application. Par conséquent, en règle générale, il convient que les données de mode de défaillance utilisées reflètent l'application pratique des composants. Un moyen précis de déterminer les "modes de défaillance" d'un élément de sous-système consiste à procéder à une AMDE. Si aucune connaissance ou information spécifique ou suffisante n'est disponible concernant les modes de défaillance, 50 % des défaillances peuvent être estimées dangereuses.

7.3.4.2 Relations entre les différents paramètres

Par hypothèse, les taux de défaillance (λ) des éléments de sous-système sont constants. Les équations de base suivantes peuvent être utilisées:

$$\lambda = \frac{1}{MTTF} \quad (3)$$

$$\lambda_D = \frac{1}{MTTF_D} \quad (4)$$

NOTE 1 Pour les besoins du calcul, la durée moyenne de fonctionnement avant défaillance ($MTTF$) peut être par hypothèse égale à la moyenne des temps de bon fonctionnement ($MTBF$).

La $MTTF$ et la $MTTF_D$ sont la plupart du temps indiquées en années [a]. Les valeurs de λ sont souvent indiquées en FIT ($FIT = failure\ in\ time$ - défaillance dans le temps) où 1 FIT signifie une défaillance de 10^9 heures.

$$1FIT = 1 \times 10^{-9} h^{-1} \quad (5)$$

Une année est à peu près égale à 8 760 heures. Par conséquent, une valeur $MTTF$ peut être convertie en valeur λ .

$$\lambda = \frac{1}{MTTF \times 8\,760 \frac{h}{a}} \quad (6)$$

NOTE 2 Exemple, $MTTF = 1000a$:

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

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

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

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

$$\lambda_{\text{exemple}} = 114,155 \text{ FIT}$$

Pour les composants pneumatiques, mécaniques et électromagnétiques (soupapes pneumatiques, relais, contacteurs, interrupteurs de position, cames d'interrupteurs de position, etc.), il peut être difficile de calculer la durée moyenne de fonctionnement avant défaillance dangereuse ($MTTF_D$) des composants, qui est donnée en années. Généralement, les fabricants de ces types de composants ne donnent que le nombre de cycles moyen jusqu'à 10 % de la défaillance dangereuse du composant (B_{10D}). Le présent Article 7 donne une méthode de calcul d'une $MTTF_D$ des composants à l'aide de la valeur de B_{10D} donnée par le fabricant étroitement liée aux cycles dépendant de l'application.

NOTE 3 Les composants hydrauliques sont le plus souvent caractérisés par la $MTTF_D$.

Si les principes de sécurité de base et les principes éprouvés sont satisfaits, la valeur $MTTF_D$ d'un seul composant pneumatique, électromagnétique ou mécanique peut être estimée.

Il convient que le nombre de cycles moyen jusqu'à 10 % des défaillances dangereuses de composants (B_{10D}) soit déterminé par le fabricant conformément aux normes de produits correspondantes pour les méthodes d'essai (l'IEC 60947-5-1, l'ISO 19973, l'IEC 61810, par exemple). Les modes de défaillance dangereuse du composant doivent être définis (en restant à une position de fin ou en modifiant les temps de commutation, par exemple). Si tous les composants ne font pas l'objet d'une anomalie dangereuse pendant les essais (cinq anomalies dangereuses sur sept composants soumis à l'essai, par exemple), il convient de procéder à une analyse tenant compte des composants qui n'ont pas fait l'objet d'une anomalie dangereuse.

Avec B_{10D} et n_{op} , le nombre moyen d'opérations annuelles, la $MTTF_D$ des composants peut être calculée comme suit

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

où

$$n_{op} = \frac{d_{op} \times h_{op} \times 3600 \frac{s}{h}}{t_{cycle}} \quad (8)$$

et avec les hypothèses suivantes formulées sur l'application du composant:

h_{op} est le fonctionnement moyen, en heures par jour;

d_{op} est le fonctionnement moyen, en jours par année;

t_{cycle} est la durée moyenne entre le début de deux cycles successifs du composant. (commutation d'une soupape, par exemple) en secondes par cycle.

En ce qui concerne le taux de défaillance λ , la relation suivante peut être exprimée sous la forme

$$\lambda_D = \frac{0,1 C}{B_{10D}} = \frac{0,1 n_{op}}{B_{10D} \times 8760 \frac{h}{a}} \quad (9)$$

où C ($C = n_{op} / 8760$) est le cycle de fonctionnement ou le fonctionnement moyen par heure.

La relation entre B_{10D} , B_{10} et le rapport de défaillance dangereuse (RDF) est

$$B_{10D} = \frac{B_{10}}{\text{rapport de défaillance dangereuse}} \quad (10)$$

La durée de fonctionnement utile du composant est limitée à T_{10D} , la durée moyenne jusqu'à 10 % des défaillances dangereuses des composants:

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

NOTE 4 Pour les systèmes électroniques, la loi exponentielle est applicable. Pour les systèmes non électroniques, la loi exponentielle n'est pas applicable. La loi de Weibull (voir également l'IEC 61649) est plus appropriée, mais les paramètres et calculs sont difficiles à appliquer. Toutefois, si la loi exponentielle est utilisée pour les composants non électroniques dans les limites de T_{10D} , les résultats des calculs sont pessimistes et la formule avec $1-e^{-\lambda t}$ peut être appliquée en tant que méthode simplifiée.

Si le rapport de défaillance dangereuse est estimé inférieur à 0,5 (50 % de défaillances dangereuses), la durée de fonctionnement utile du composant est limitée à deux fois T_{10} .

Le rapport de défaillance dangereuse est estimé à 0,5 (50 % de défaillances dangereuses) si aucune autre information (norme de produit, par exemple) n'est disponible.

7.4 Contraintes architecturales d'un sous-système

7.4.1 Généralités

Dans le contexte de l'intégrité de sécurité du matériel, le niveau d'intégrité de sécurité le plus élevé qui peut être revendiqué pour un SCS est limité par les tolérances aux anomalies du matériel (HFT) et les proportions de défaillances en sécurité (*SFF*) des sous-systèmes qui réalisent cette fonction de sécurité. Le Tableau 6 spécifie le niveau d'intégrité de sécurité le plus élevé qui peut être revendiqué pour un SCS qui utilise un sous-système prenant en compte la tolérance aux anomalies du matériel et la proportion de défaillances en sécurité de ce sous-système. Les contraintes architecturales données dans le Tableau 6 doivent s'appliquer à chaque sous-système développé selon l'Article 7. Prenant en compte ces contraintes architecturales:

- a) une tolérance aux anomalies du matériel N signifie que $N+1$ anomalies peuvent provoquer la perte de la fonction de sécurité. Lors de la détermination de la tolérance aux anomalies du matériel, aucune autre mesure pouvant maîtriser l'effet des anomalies, telle que les diagnostics, n'est prise en compte; et
- b) lorsqu'une anomalie donne directement lieu à l'apparition d'une ou de plusieurs anomalies subséquentes, celles-ci doivent être considérées comme un premier défaut;
- c) lors de la détermination de la tolérance aux anomalies du matériel, certaines anomalies peuvent être exclues, à condition que leur probabilité d'occurrence soit très faible par rapport aux exigences d'intégrité de sécurité du sous-système (voir 7.3.3.3).

Un sous-système comprenant uniquement un seul élément de sous-système doit satisfaire aux exigences du Tableau 4. En particulier, pour un élément de sous-système de niveau SIL 3 ayant une tolérance aux anomalies du matériel de zéro (HFT 0), une *SFF* supérieure à 99 % doit alors être réalisée par une fonction de diagnostic d'un SCS.

Si au moins deux sous-systèmes types sont combinés en un sous-système redondant, les contraintes architecturales du sous-système combiné peuvent être déterminées. Cela peut être réalisé en prenant le sous-système dont le niveau SIL est le plus élevé selon les contraintes architecturales, et en recherchant le SIL correspondant dans le Tableau 6, colonne HFT0. Cela renvoie la plage de *SFF* applicable. Le SIL du sous-système combiné doit être déduit en augmentant la HFT d'un (1) dans la plage de *SFF* selon l'IEC 61508-2:2010, 7.4.4.2.4.

NOTE 2 Cette procédure s'applique uniquement aux sous-systèmes combinés avec un niveau SIL défini.

Tableau 6 – Contraintes architecturales sur un sous-système: SIL maximal pouvant être revendiqué pour un SCS utilisant ce sous-système

Proportion de défaillances en sécurité (SFF)	Tolérance aux anomalies du matériel (HFT) (voir NOTE 1)		
	0	1	2
< 60 %	Non admis (pour les exceptions, voir NOTE 3)	SIL 1	SIL 2
60 % à < 90 %	SIL 1	SIL 2	SIL 3
90 % à < 99 %	SIL 2	SIL 3	SIL 3 (voir NOTE 2)
≥ 99 %	SIL 3	SIL 3 (voir NOTE 2)	SIL 3 (voir NOTE 2)

NOTE 1 Une tolérance aux anomalies du matériel N signifie que $N + 1$ anomalies peuvent provoquer une perte de la fonction de sécurité.

NOTE 2 SIL 4 n'est pas pris en considération dans le présent document. Pour le SIL 4, voir l'IEC 61508-1.

NOTE 3 Se référer à 7.4.3.2 lorsque les sous-systèmes dont la proportion de défaillances en sécurité est inférieure à 60 %, ne présentant aucune tolérance aux anomalies du matériel et qui utilisent des composants éprouvés peuvent être considérés comme étant de niveau SIL 1; ou s'y référer pour les sous-systèmes dont les exclusions d'anomalies ont été appliquées aux anomalies qui peuvent donner lieu à une défaillance dangereuse.

NOTE 4 Dans l'IEC 62061:2015, le SIL maximal qui peut être revendiqué a été appelé SILCL.

NOTE 5 Voir 7.3.3.3 pour la limitation de SIL lors de l'application de l'exclusion d'anomalie.

NOTE 6 Pour HFT0 à $SFF \geq 99\%$, cela est uniquement possible en cas de surveillance continue du bon fonctionnement de l'élément. En règle générale, une technologie électronique est nécessaire à cet effet.

7.4.2 Estimation de la proportion de défaillances en sécurité (SFF)

Pour estimer la *SFF*, une analyse (par exemple analyse par arbre de panne, analyse des modes de défaillance et de leurs effets) de chaque sous-système doit être réalisée afin de déterminer toutes les anomalies appropriées et leurs modes de défaillance correspondants. Qu'une défaillance soit une défaillance en sécurité ou une défaillance dangereuse dépend du SCS et de la fonction de sécurité prévue, y compris la fonction réaction à l'anomalie (7.4.3). La probabilité de chaque mode de défaillance doit être déterminée sur la base de la probabilité des anomalies associées en prenant en compte l'usage prévu, et peut être déduite de sources telles que:

- les données de taux de défaillance suffisamment fiables issues du retour d'expérience du fabricant et appropriées à l'usage prévu;
- les données de défaillance de composants provenant d'une origine industrielle reconnue et appropriées à l'usage prévu;
- les données de taux de défaillance déduits des résultats des essais et des analyses.

NOTE 1 Des informations sur les taux de défaillance du composant électrique/électronique peuvent être consultées dans différentes sources, parmi lesquelles: MIL-HDBK 217F, MIL-HDBK 217F (Annexe A), SN 29500 Parties 7 et 11, IEC 61709, FMD-2016, OREDA Handbook, EXIDA Safety Equipment Reliability Handbook et EXIDA Electrical & Mechanical Component Reliability Handbook.

NOTE 2 Les données relatives au taux de défaillance peuvent être fournies par les fabricants.

NOTE 3 Certaines normes de composants fournissent des données pertinentes (par exemple, l'Annexe K de l'IEC 60947-4-1:2018).

NOTE 4 Des listes d'anomalies à prendre en considération pour les technologies mécaniques, pneumatiques, hydrauliques et électriques sont données aux Annexes A, B, C et D de l'ISO 13849-2:2012.

En général, la *SFF* peut être calculée comme suit:

$$SFF = \frac{\sum \lambda_s + \sum \lambda_{DD}}{\sum \lambda_s + \sum \lambda_D} \quad (12)$$

où

λ_s est le taux de défaillance en sécurité,

$\sum \lambda_s + \sum \lambda_D$ est le taux global des défaillances,

λ_{DD} est le taux de défaillance dangereuse qui est détecté par les fonctions de diagnostic,

λ_D est le taux de défaillance dangereuse.

La défaillance d'un élément qui joue un rôle dans la mise en œuvre de la fonction de sécurité, mais qui n'a aucun effet direct (défavorable) sur la fonction de sécurité, est appelée "défaillance n'ayant aucun effet" et n'est pas considérée comme une défaillance en sécurité (λ_s). Par conséquent, elle ne doit pas être utilisée pour calculer la *SFF*.

En général, pour les composants non électroniques, λ_s est par hypothèse égale à 0 ou est négligeable car, dans la plupart des cas, elle est insignifiante par rapport à λ_D . Dans ce cas, la simplification suivante peut être appliquée (voir également l'exemple à l'Article B.4):

$$SFF = \frac{\sum \lambda_s + \sum \lambda_{DD}}{\sum \lambda_s + \sum \lambda_D} = \frac{\sum \lambda_{DD}}{\sum \lambda_D} \quad (13)$$

EXEMPLE 2 Si la tolérance aux anomalies du matériel d'un sous-système est égale à 0, la *SFF* devient

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

où DC_1 est la couverture de diagnostic de l'élément de sous-système 1.

EXEMPLE 3 Si la tolérance aux anomalies du matériel d'un sous-système est égale à 1, la *SFF* devient

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

où DC_1 et DC_2 sont les couvertures de diagnostic des éléments de sous-système 1 et 2 respectivement (voir également 7.4.2 pour la relation entre λ et *MTTF*).

7.4.3 Comportement (du SCS) lors de la détection d'une anomalie dans un sous-système

7.4.3.1 Généralités

La détection d'une anomalie dangereuse dans un quelconque sous-système ayant une tolérance aux anomalies du matériel supérieure à zéro doit déclencher l'exécution de la fonction réaction à l'anomalie spécifiée.

La spécification peut permettre l'isolement de la partie du sous-système présentant l'anomalie afin de permettre la poursuite en sécurité de l'exploitation de la machine, pendant que la partie

présentant une anomalie est réparée. Dans ce cas, si la partie présentant une anomalie n'est pas réparée durant le temps estimé maximal, pris comme hypothèse dans le calcul de la *PFH*, alors une seconde réaction à l'anomalie doit avoir lieu afin d'atteindre un état de sécurité.

Si le SCS est conçu pour être réparé en ligne, l'isolement de la partie présentant une anomalie ne doit être appliqué que si cela n'accroît pas la *PFH* du SCS au-dessus de celle spécifiée dans la SRS.

Tant que le fonctionnement se poursuit et que la tolérance aux anomalies du matériel est réduite à zéro, les exigences de 7.4.3.2 s'appliquent.

7.4.3.2 Fonction réaction à l'anomalie

Lorsqu'une fonction de diagnostic est nécessaire à l'obtention de la *PFH* exigée ou de la proportion de défaillances en sécurité et que le sous-système présente une tolérance aux anomalies du matériel de zéro, alors

- la somme de l'intervalle entre essais de diagnostic et de la durée d'exécution de la fonction réaction à l'anomalie spécifiée pour atteindre ou maintenir un état de sécurité doit être inférieure au temps de sécurité du processus (voir l'ISO 13855, par exemple); ou,
- lors du fonctionnement en mode sollicitation élevée, le rapport du taux d'essai de diagnostic sur le taux de sollicitation doit être supérieur ou égal à 100.

Lorsque l'exécution d'une fonction réaction à l'anomalie, partie d'une SCS spécifiée SIL 3, a entraîné l'arrêt de la machine, le fonctionnement normal ultérieur de la machine à l'aide du SCS (permettant le redémarrage de la machine, par exemple) ne doit pas être possible tant que l'anomalie n'a pas été réparée ou corrigée. Pour un SCS avec un fonctionnement de la sécurité spécifié inférieur à SIL 3, le comportement de la machine après l'exécution d'une fonction réaction à l'anomalie (redémarrant le fonctionnement normal, par exemple) doit dépendre de la spécification des fonctions réactions aux anomalies appropriées (voir 5.2.2).

7.4.3.3 Couverture du diagnostic (*DC*)

La couverture du diagnostic (*DC*) peut être calculée comme la proportion de défaillances dangereuses à l'aide de l'équation suivante:

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

où λ_{DD} est le taux de défaillances dangereuses détectées pour le matériel et λ_D est le taux de défaillances dangereuses pour le matériel.

Dans la plupart des cas, pour estimer la *DC*, l'analyse des modes de défaillance et de leurs effets (AMDE – voir l'IEC 60812), l'analyse des modes de défaillance, de leurs effets et du diagnostic (AMDED) ou des méthodes équivalentes peuvent être utilisées. Dans ce cas, il convient de prendre en considération toutes les anomalies et/ou tous les modes de défaillance concernés.

Pour une approche simplifiée de l'estimation de la couverture du diagnostic, voir l'Annex D.

NOTE L'Annexe C de l'IEC 61508-2:2010 donne de plus amples informations.

7.4.4 Réalisation des fonctions de diagnostic

Chaque sous-système doit être équipé des fonctions de diagnostic associées nécessaires pour satisfaire aux exigences en matière de contraintes architecturales et de *PFH*.

Les fonctions de diagnostic sont considérées comme des fonctions séparées qui peuvent avoir une structure différente de celle de la fonction de sécurité et peuvent être réalisées par

- le même sous-système qui exige des diagnostics; ou
- d'autres sous-systèmes du SCS; ou
- des sous-systèmes du SCS n'effectuant pas la fonction de sécurité.

Les fonctions de diagnostic doivent satisfaire aux exigences suivantes:

- les exigences applicables en matière d'évitement des défaillances systématiques; et
- les exigences applicables en matière de maîtrise des défaillances systématiques.

NOTE 1 Les contraintes temporelles applicables aux essais du sous-système qui exécute une fonction de diagnostic peuvent être différentes de celles applicables aux fonctions de sécurité.

NOTE 2 La nécessité de contrôler la fonction de diagnostic peut dépendre, par exemple, du niveau d'intégrité de sécurité, du taux de sollicitation, de la technologie utilisée et des capacités spécifiques à l'application.

Une description claire de la ou des fonctions de diagnostic du SCS, de leur détection/réaction vis-à-vis des défaillances, ainsi qu'une analyse de leur contribution à l'intégrité de sécurité des fonctions de sécurité associées doivent être fournies.

Pour appliquer l'approche simplifiée du présent document et estimer la *PFH* des sous-systèmes, les dispositions suivantes doivent s'appliquer:

La ou les fonctions de diagnostic du SCS doivent au minimum être mises en œuvre de sorte que la *PFH* et l'intégrité de sécurité systématique soient identiques à celles spécifiées pour la ou les fonctions de sécurité correspondantes,

ou

si l'amplitude de la *PFH* est plus importante que celle spécifiée pour la fonction de sécurité, un essai doit être réalisé pour déterminer si la ou les fonctions de diagnostic restent opérationnelles. Un essai de la ou des fonctions de diagnostic doit être réalisé au moins 10 fois à intervalles réguliers pendant l'intervalle d'essai périodique du sous-système.

NOTE 3 Les contraintes architecturales sur l'intégrité de sécurité du matériel ne s'appliquent pas à la réalisation de la ou des fonctions de diagnostic.

NOTE 4 Un essai de la ou des fonctions de diagnostic est prévu pour couvrir autant que possible 100 % des parties mettant en œuvre la ou les fonctions de diagnostic.

NOTE 5 Lorsqu'une fonction de diagnostic est réalisée par l'unité logique du SCS, il peut ne pas être nécessaire de réaliser un essai particulier de la fonction de diagnostic puisque sa défaillance peut être révélée par une défaillance de la fonction de sécurité.

NOTE 6 Un essai peut être réalisé soit par des moyens externes (un matériel d'essai, par exemple) ou par des vérifications dynamiques internes (intégrées dans l'unité logique, par exemple) du SCS.

7.5 Architectures de conception du sous-système

7.5.1 Généralités

L'architecture d'un sous-système décrit dans le présent paragraphe peut être utilisée pour évaluer les contraintes architecturales et estimer la *PFH* (voir l'Annex H).

NOTE Les figures en 7.5 représentent une vue logique des architectures de sous-système et ne visent pas à représenter des schémas de connexion physique particuliers. Une tolérance aux anomalies du matériel de 1 est représentée par des éléments de sous-système parallèles, mais les connexions physiques dépendent de l'application du sous-système.

7.5.2 Architectures de sous-système simple

7.5.2.1 Architecture A d'un sous-système simple: simple canal sans fonction de diagnostic

Dans cette architecture (voir la Figure 8), une défaillance dangereuse d'un élément de sous-système entraîne une défaillance de la fonction de sécurité. Cette architecture correspond à une tolérance aux anomalies du matériel de 0.

En mode de fonctionnement à forte sollicitation ou continu, l'architecture de type A ne doit pas s'appuyer sur un essai périodique.

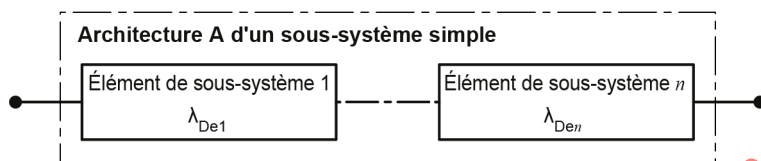


Figure 8 – Représentation logique d'un sous-système de type A

7.5.2.2 Architecture B d'un sous-système simple: double canal sans fonction de diagnostic

Cette architecture (voir la Figure 9) est telle qu'une défaillance unique d'un élément de sous-système n'entraîne pas une perte de la fonction de sécurité. Cette architecture correspond à une tolérance aux anomalies du matériel de 1.

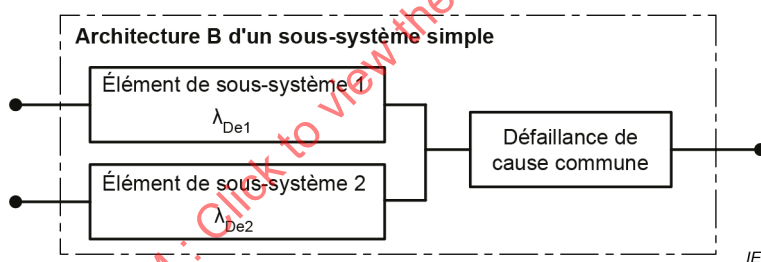


Figure 9 – Représentation logique d'un sous-système de type B

7.5.2.3 Architecture C d'un sous-système simple: simple canal avec fonction de diagnostic

Dans cette architecture (voir la Figure 10), une anomalie dangereuse non détectée d'un élément de sous-système conduit à une défaillance dangereuse de la fonction de sécurité.

Lorsqu'une anomalie d'un élément de sous-système est détectée, la ou les fonctions de diagnostic déclenchent une fonction réaction à l'anomalie (voir 7.4.3). Cette architecture correspond à une tolérance aux anomalies du matériel de 0.

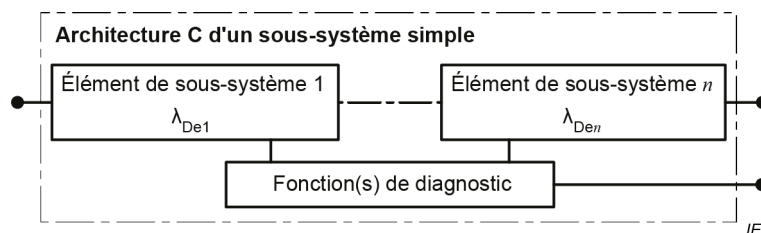


Figure 10 – Représentation logique d'un sous-système de type C

7.5.2.4 Architecture D d'un sous-système simple: double canal avec fonction(s) de diagnostic

Cette architecture (voir la Figure 11) est telle qu'une défaillance unique d'un élément de sous-système n'entraîne pas une perte de la fonction de sécurité. Lorsqu'une anomalie d'un élément de sous-système est détectée, la ou les fonctions de diagnostic déclenchent une fonction réaction à l'anomalie (voir 7.4.3). Cette architecture correspond à une tolérance aux anomalies du matériel de 1.

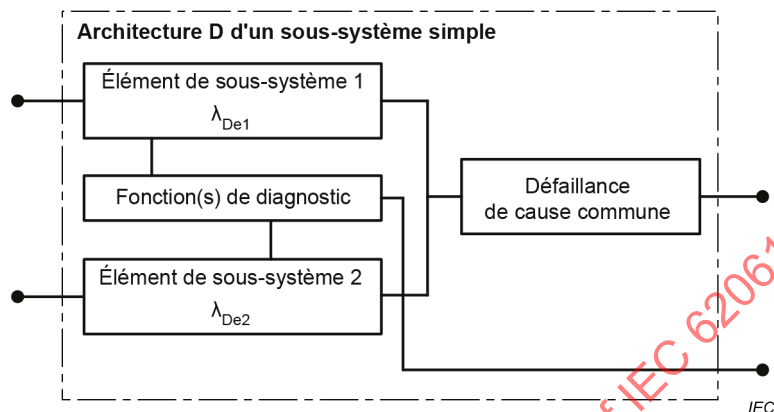


Figure 11 – Représentation logique d'un sous-système de type D

7.5.3 Exigences de base

Comme le représente le Tableau 7, les exigences de base qui dépendent des contraintes architecturales et des architectures de sous-système simple doivent être appliquées.

Tableau 7 – Aperçu des exigences de base et de l'interrelation avec les architectures de sous-système simple

Exigences de base	Tolérance aux anomalies du matériel (HFT)				Commentaires / Exemples
	0		1		
	SFF		SFF		
	< 60 %	≥ 60 %	< 60 %	≥ 60 %	
Principes de sécurité de base	O	O	O	O	Utilisation de matériaux adaptés ISO 13849-2:2012, Annexe A à Annexe D
Principes de sécurité éprouvés	O	O	O	O	Contacts mécaniquement liés et contacts à action d'ouverture directe ISO 13849-2:2012, Annexe A à Annexe D
Composants éprouvés	O	--	--	--	Contacteur (IEC 60947-4-1) ISO 13849-2:2012, Annexe A à Annexe D
CCF	non pertinent	O	O	O	
Type d'architecture de sous-système simple	A	C	B	D	

O = obligatoire; -- = pas d'exigence

NOTE Le Tableau 6 relatif aux contraintes architecturales est toujours applicable.

7.6 Fréquence moyenne de défaillance dangereuse par heure (PFH) des sous-systèmes

7.6.1 Généralités

Les paramètres suivants doivent être définis pour pouvoir déterminer la *PFH*:

- architecture de sous-système (voir 7.5);
- couverture du diagnostic et intervalles d'essai (voir 7.4.3 et 7.4.4);
- CCF (voir l'Annex E);
- λ_D ou $MTTF_D$ des éléments de sous-système (voir 7.3.4);
- durée de fonctionnement utile.

NOTE Étant donné que la durée de vie classique d'une machine est d'environ 20 ans, une durée de fonctionnement utile de 20 ans est préférentielle. Il s'agit de clarifier la période d'usage maximale du sous-système. Pour les composants présentant des caractéristiques d'usure, la durée de fonctionnement utile peut être limitée par T_{10D} .

7.6.2 Méthodes d'estimation de la *PFH* d'un sous-système

L'une des méthodes suivantes de l'Annex H peut être utilisée pour calculer la *PFH* dans le cadre d'une approche simplifiée:

- approche du tableau d'allocation (voir l'Article H.1);
- formules (voir l'Article 0).

Une modélisation reposant, par exemple, sur l'analyse par arbre de pannes (voir B.6.6.5 de l'IEC 61508-7:2010 et l'IEC 61025), sur les modèles de Markov (voir B.6.6.6, C.6.4 de

l'IEC 61508-7:2010 et l'IEC 61165) ou sur les schémas fonctionnels de fiabilité (voir C.6.4 de l'IEC 61508-7:2010) est toujours possible.

7.6.3 Approche simplifiée pour l'estimation de la contribution des défaillances de cause commune (CCF)

La connaissance de la sensibilité d'un sous-système à la CCF est exigée pour contribuer à l'estimation de la *PFH* d'un sous-système.

La probabilité d'occurrence de la CCF dépend habituellement d'une combinaison de technologie, d'architecture, d'application et d'environnement. L'utilisation de l'Annex E est pertinente pour éviter de nombreux types de CCF.

8 Logiciels

8.1 Généralités

Toutes les activités de cycle de vie des logiciels d'application relatifs à la sécurité doivent consister à éviter les anomalies introduites lors du cycle de vie du logiciel. Les exigences suivantes ont pour principal objet de produire des logiciels lisibles, compréhensibles, testables, actualisables et corrects.

Lorsque le logiciel réalise à la fois des fonctions relatives à la sécurité et des fonctions non relatives à la sécurité, il doit alors être traité comme étant relatif à la sécurité, sauf si une indépendance suffisante entre les fonctions peut être démontrée dans la conception. Dans toute la mesure du possible, il est donc préférable de séparer les fonctions non relatives à la sécurité (les fonctions de base de la machine, par exemple) des fonctions de sécurité.

Le présent document doit uniquement être utilisé pour les logiciels d'application qui fonctionnent sur une plateforme type conforme à l'IEC 61508 ou à d'autres normes de sécurité fonctionnelle liées à l'IEC 61508 (par exemple, l'IEC 61131-6).

NOTE Dans la suite du présent article le logiciel d'application est également désigné par le terme "logiciel".

8.2 Définition des niveaux logiciels

Le présent document décrit trois différents niveaux de logiciels d'application (voir le Tableau 8).

Tableau 8 – Différents niveaux de logiciels d'application

Niveau logiciel	Principe essentiel	Sous-principe	Exemple
1	Plateforme (combinaison de matériel et de logiciels) type conforme à l'IEC 61508 ou à d'autres normes de sécurité fonctionnelle liées à l'IEC 61508 (par exemple, l'IEC 61131-6). Logiciel d'application utilisant le langage de variabilité limitée (LVL).	Logiciel d'application conforme au présent document.	Contrôleur logique programmable (PLC) de sécurité avec LVL ou relais programmable de sécurité
2	Plateforme (combinaison de matériel et de logiciels) type conforme à l'IEC 61508 ou à d'autres normes de sécurité fonctionnelle liées à l'IEC 61508 (par exemple, l'IEC 61131-6).	Logiciel d'application conforme au présent document.	PLC de sécurité avec FVL (FVL conforme au présent document.)
3	Logiciel d'application utilisant un autre langage que le langage de variabilité limitée (LVL).	Logiciel d'application conforme à l'IEC 61508-3.	PLC de sécurité avec LVL ou langage de variabilité totale (FVL) (FVL conforme à l'IEC 61508)

NOTE 1 Le niveau logiciel 2 est introduit pour prendre en charge le langage de variabilité totale, mais se limite à SIL 2. Quant aux logiciels d'application conformes à SIL 3 (c'est-à-dire, au niveau logiciel 3), ils sont satisfont aux exigences de l'IEC 61508-3.

NOTE 2 Pour les autres types de plateformes, aucune exigence n'est formulée dans le présent document. L'IEC 61508-2 et l'IEC 61508-3 décrivent la manière de traiter ces systèmes.

Le langage de programmation (jeu d'instructions) à utiliser pour le logiciel d'application doit relever du domaine de sécurité de la plateforme, préconçue selon l'IEC 61508 ou d'autres normes de sécurité fonctionnelle liées à l'IEC 61508 (par exemple, l'IEC 61131-6).

Le langage de programmation à utiliser et les outils (du cycle de vie de développement des logiciels) doit être adapté à la création de logiciels d'application liés à la sécurité sur la plateforme; voir 8.4.1.3.

Dans ce contexte, la plateforme décrite dans le Tableau 8 doit exiger uniquement les logiciels d'application pour exécuter sa fonctionnalité relative à la sécurité.

NOTE Par exemple, les éléments tels que les systèmes sur puce ou les cartes de microcontrôleur ne sont pas des plateformes dans ce sens.

Le niveau logiciel 3 n'est plus décrit dans le présent document, car il est couvert par l'application correcte des parties respectives de l'IEC 61508. Un niveau élevé de compétence est exigé en matière de conception en fonction du niveau logiciel 2 ou 3. Les facteurs qui rendent l'utilisation de l'IEC 61508-3 (niveau logiciel 3) plus appropriée que celle du présent document (niveau logiciel 2) sont les suivants:

- degré élevé de complexité de la ou des fonctions de sécurité;
- grand nombre de fonctions de sécurité;
- projet de grande taille.

Les exigences concernant le cycle de vie de sécurité du logiciel pour les différents niveaux logiciels sont détaillées dans les paragraphes ci-dessous:

- Niveau logiciel 1: voir 8.3;
- Niveau logiciel 2: voir 8.4.

8.3 Niveau logiciel 1

8.3.1 Cycle de vie de sécurité du logiciel – Niveau logiciel 1

8.3.1.1 Niveau SIL maximal atteignable – Niveau logiciel 1

Le niveau SIL maximal atteignable pour le niveau logiciel 1 est SIL 3.

8.3.1.2 Modèle de cycle de vie de sécurité du logiciel – Niveau logiciel 1

Un modèle de cycle de vie de sécurité du logiciel résolu en différentes phases doit être utilisé (modèle en V, par exemple), y compris les activités de gestion et de documentation permettant d'obtenir le niveau de sécurité exigé.

Un modèle de cycle de vie du logiciel peut être utilisé à condition que tous les objectifs et toutes les exigences du présent 8.3 soient satisfaits. Le logiciel relatif à la sécurité doit être validé comme cela est indiqué en 9.5.4.

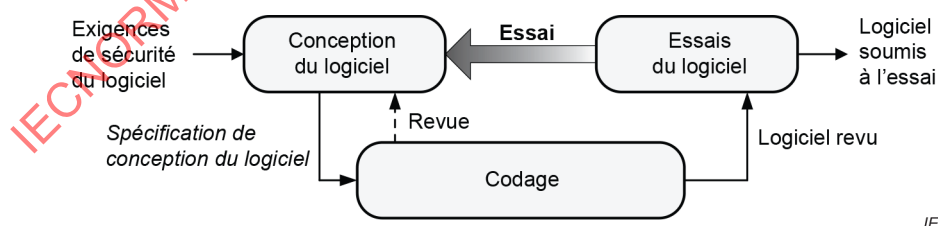
Le niveau logiciel 1 présente une complexité réduite en raison de l'utilisation de modules matériels et logiciels relatifs à la sécurité types. Par conséquent, le modèle en V simplifié de la Figure 12 s'applique. La conception de modules logiciels personnalisés ou créés automatiquement peut s'avérer nécessaire (si les modules de bibliothèque fournis par le fabricant de composants se révèlent inadéquats ou inadaptés, par exemple). La conception de modules logiciels personnalisés par le concepteur est une activité supplémentaire qui doit être réalisée selon le modèle en V de la Figure 13.

NOTE 1 Un module logiciel (abrégié en module) est une unité fonctionnelle du logiciel, qui est en général uniquement accessible par l'intermédiaire de son interface d'entrée et de sortie. Il est réutilisable et facilite le développement logiciel modulaire. Les modules logiciels font souvent partie d'une bibliothèque. En programmation PLC, les modules logiciels sont des fonctions ou des blocs fonctionnels.

NOTE 2 Le modèle en V est un modèle statique utilisé pour structurer la conception du logiciel en petites parties. Il n'introduit aucune séquence de création de spécifications ni de mise en œuvre. Le côté gauche représente les exigences, c'est-à-dire les choses à atteindre. Le côté droit décrit les essais du logiciel.

NOTE 3 Sur le côté gauche du modèle en V, chaque phase est examinée. Il s'agit de vérifier la sortie d'une phase dans le modèle en V par rapport aux exigences de l'entrée de la même phase. La flèche "Examiner" représente la première étape de la vérification du logiciel. De plus amples informations sur le niveau d'indépendance de l'examen et de l'essai ou de la vérification sont disponibles dans l'Annex J.

NOTE 4 Le cycle de vie est assorti de techniques et processus de gestion de projet appropriés pour la taille et le domaine d'application du projet.



IEC

Figure 12 – Modèle en V pour le niveau logiciel 1

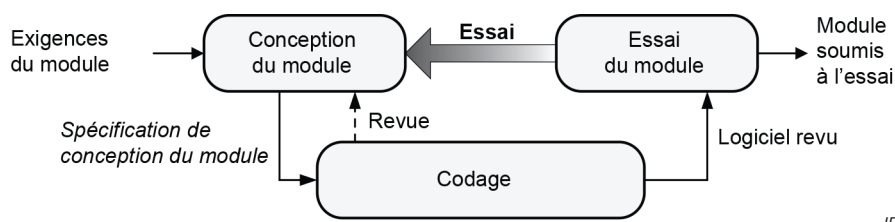


Figure 13 – Modèle en V pour les modules logiciels personnalisés par le concepteur pour le niveau logiciel 1

NOTE 5 Dans les modèles en V, la flèche "Essai" représente les résultats des cas d'essai en fonction de la spécification et, de plus, la nécessité d'exigences et de spécifications plus précises concernant le cas d'essai.

NOTE 6 Le résultat de la Figure 13 est une entrée au codage de la Figure 12.

8.3.1.3 Utilisation des outils – Niveau logiciel 1

Les outils doivent être utilisés selon les instructions du fabricant correspondant du ou des systèmes relatifs à la sécurité (PLC, matériel de protection électrosensible, par exemple).

8.3.2 Conception du logiciel – Niveau logiciel 1

8.3.2.1 Niveau logiciel 1 – Généralités

Lorsqu'une partie quelconque d'un SCS qui met en œuvre une fonction de sécurité doit utiliser le logiciel, la spécification des exigences de sécurité du logiciel doit être développée, documentée et gérée tout au long du cycle de vie du SCS.

La spécification des exigences de sécurité du logiciel doit être développée pour chaque sous-système sur la base de la spécification et de l'architecture du SCS.

8.3.2.2 Exigences de sécurité du logiciel – Niveau logiciel 1

Pour prendre en charge le processus de conception du logiciel, les informations suivantes doivent être prises en considération:

- spécification de la ou des fonctions de sécurité (voir 5.2);
- configuration ou architecture du SCS (architecture du matériel, schéma de câblage, entrées et sorties relatives à la sécurité, par exemple);
- exigences de temps de réponse;
- interfaces et commandes de l'opérateur (interrupteurs, joysticks, sélecteur de mode, cadrans, dispositifs de commande tactiles, claviers, etc.);
- modes de fonctionnement adaptés de la machine;
- exigences relatives aux diagnostics du matériel, y compris les caractéristiques des capteurs, des actionneurs finaux, etc.;
- effets des tolérances mécaniques (des capteurs et/ou de leurs contreparties sensibles, par exemple);
- lignes directrices en matière de codage.

8.3.2.3 Spécification de conception du logiciel – Niveau logiciel 1

La spécification de conception du logiciel doit être déduite des exigences de sécurité du logiciel du SCS.

La spécification de conception du logiciel doit être:

- structurée, examinable, testable, compréhensible, actualisable et manœuvrable;
- développée pour chaque sous-système sur la base de la spécification et de l'architecture du SCS;
- suffisamment détaillée pour permettre la conception et la mise en œuvre du SCS afin d'atteindre le niveau de sécurité (SIL) exigé, ainsi que la vérification et les essais.
- traçable par rapport à la spécification des exigences de sécurité du logiciel du SCS. Cela signifie que la spécification est par nature compréhensible de sorte qu'une autre personne (n'étant pas spécialiste en matière de logiciels, par exemple) puisse vérifier si elle correspond aux exigences de sécurité du logiciel des fonctions de sécurité définies dans l'appréciation du risque;
- exempte de toute terminologie ambiguë et descriptions non pertinentes.

Il doit être possible d'associer directement les entrées de la spécification de conception du logiciel aux sorties souhaitées, et inversement. Le cas échéant, des méthodes semi-formelles aisément lisibles (des tables de cause à effet, des tableaux ou schémas logiques, des blocs fonctionnels ou des diagrammes de séquence, par exemple) doivent être utilisées dans la documentation.

NOTE 1 Cela dépend, le cas échéant, du nombre de fonctions de sécurité impliquées dans le programme. À chaque fois que le nombre total de fonctions à l'intérieur du programme dépasse 3, il est considéré comme étant approprié.

Les éléments suivants doivent être précisés dans la spécification de conception du logiciel:

- a) la logique des fonctions de sécurité, y compris les entrées et sorties relatives à la sécurité et les diagnostics corrects des anomalies détectées. Les méthodes possibles incluent, entre autres, la table de cause à effet, une description écrite ou des blocs fonctionnels;

NOTE 2 Les anomalies peuvent également être détectées par le matériel (incohérence de signal détectée par la carte d'entrée, par exemple).

- b) les cas d'essai, qui incluent:

- la ou les valeurs d'entrée particulières pour lesquelles l'essai est réalisé et les résultats d'essai prévus, y compris les critères de réussite/d'échec;
- l'insertion ou l'injection ou les injections d'anomalie.

NOTE 3 Pour les fonctions simples, le ou les cas d'essai peuvent être donnés de manière implicite par la spécification de la fonction de sécurité.

- c) les fonctions de diagnostic des dispositifs d'entrée (les éléments de détection et les interrupteurs, par exemple) et les éléments de commande finale (les solénoïdes, les relais ou les contacteurs, par exemple);
- d) les fonctions qui permettent à la machine d'atteindre ou de maintenir un état de sécurité;
- e) les fonctions liées à la détection, la signalisation et le traitement des anomalies;
- f) les fonctions liées aux essais périodiques du ou des SCS en ligne et hors ligne;
- g) les fonctions qui empêchent une modification non autorisée du SCS (un mot de passe, par exemple);
- h) les interfaces vers des éléments non SCS;
- i) le temps de réponse de la fonction de sécurité.

NOTE 4 Des recommandations relatives à la documentation du logiciel sont données dans l'IEC 61508 et l'ISO/IEC/IEEE 26512.

La spécification de la conception du logiciel doit également expliquer les principaux aspects du logiciel. Les aspects principaux incluent par exemple:

- le cas échéant, l'architecture du logiciel qui définit la structure décidée en vue de satisfaire à la spécification de conception du logiciel;
- les données globales;
- les bibliothèques de données utilisées;
- les modules logiciels préexistants utilisés;
- les fonctions de diagnostic (internes, externes);
- les outils de programmation, y compris les informations qui identifient l'outil de manière unique;
- les cas d'essai et procédures d'intégration, y compris la spécification de l'environnement d'essai, le logiciel support, la description de la configuration et les procédures d'actions correctives en cas de défaillance d'un essai.

Dans la mesure du possible, il est recommandé d'utiliser des modules logiciels types dans la spécification de conception du logiciel, par exemple un module logiciel utilisé pour la fonction d'inhibition selon l'IEC 61496-1 et conçu par le fabricant de la plateforme.

Dans le cas des sous-fonctions de sécurité types (IEC 61800-5-2, par exemple), il est recommandé de faire référence à la spécification fournie par le fabricant.

Les informations présentées dans la spécification de conception du logiciel doivent être examinées et, le cas échéant, révisées de manière à assurer que les exigences de sécurité du logiciel (voir 8.3.2.2) sont correctement spécifiées.

8.3.3 Conception du module – Niveau logiciel 1

8.3.3.1 Niveau logiciel 1 – Généralités

Lorsque des modules de bibliothèque logicielle développés précédemment doivent être utilisés pour la conception, leur capacité à satisfaire à la spécification des exigences de sécurité du logiciel doit être démontrée. Les contraintes imposées par l'environnement de développement du logiciel précédent (par exemple les dépendances au système d'exploitation et au compilateur) doivent être évaluées.

8.3.3.2 Informations d'entrée – Niveau logiciel 1

Pour les modules logiciels, les informations suivantes doivent être disponibles dans les exigences du module:

- a) la description du module;
- b) l'interface du module (entrées et sorties avec les types de données et, le cas échéant, avec les plages de données);
- c) les bibliothèques de modules utilisées;
- d) les règles de codage particulières.

8.3.3.3 Spécification de conception du module – Niveau logiciel 1

La spécification de conception du module doit contenir les informations suivantes:

- a) la description de la logique (c'est-à-dire la fonctionnalité) de chaque module;
- b) les interfaces d'entrée et de sortie intégralement définies pour chaque module;
- c) les formats et les plages de valeurs des données d'entrée et de sortie et leur relation avec les modules;
- d) les cas d'essai qui doivent inclure le fonctionnement normal et normal en extérieur.

NOTE Même si les cas d'essai comprennent souvent les essais individuels des paramètres dans leurs plages spécifiées, une variation de combinaison de ces paramètres peut introduire un fonctionnement intempestif.

Ces informations doivent être examinées par rapport aux informations d'entrée (voir 8.3.3.2).

8.3.4 Codage – Niveau logiciel 1

Le logiciel doit être développé conformément aux spécifications de conception et aux règles de codage. Les règles de codage peuvent être des normes de l'industrie reconnues ou peuvent être internes au fabricant. Le code doit être examiné par rapport aux spécifications de conception et aux règles de codage.

NOTE Les règles de codage sont destinées à limiter la liberté de programmation, afin d'éviter que le code de programme ne devienne incompréhensible et de limiter la probabilité d'entrée du programme dans des états imprévus.

La sortie du codage doit être composée

- de la liste du code source (échelle, blocs fonctionnels, modèles, par exemple);
- du rapport de revue de code.

Certaines règles de codage classiques à appliquer sont, entre autres, les suivantes:

- La structure du programme est aussi facile et claire que possible.
- Il convient que la structure du programme soit telle que le flux logique commence en haut et suive la séquence effective.
- Il convient que chaque partie comporte des commentaires suffisants de manière prédéfinie.
- Il convient d'utiliser les mêmes noms pour les paramètres que ceux utilisés lors de la conception.
- Il convient que le nom représente de manière claire la fonction du paramètre.
- Il convient de disposer d'un état prédéfini.
- Il convient de limiter l'utilisation des termes définir/redéfinir pour les fonctions de sécurité.
- Il convient que les sorties de sécurité ne soient attribuées qu'une seule fois à l'intérieur d'un programme.
- Les paramètres non relatifs à la sécurité ne doivent pas être utilisés pour éviter des fonctions de sécurité.

8.3.5 Essai du module – Niveau logiciel 1

Chaque module qui n'a pas été préalablement évalué doit être soumis à l'essai en fonction des cas d'essai définis dans la conception du module par des essais fonctionnels et des essais boîte noire, boîte grise ou boîte blanche, selon le cas.

Si le module ne satisfait aux essais, une action corrective prédéfinie doit être réalisée.

Les résultats d'essai et les actions correctives doivent être documentés.

NOTE 1 Les essais fonctionnels visent à révéler les défaillances pendant les phases de spécification et de conception, et à éviter les défaillances lors de la mise en œuvre et de l'intégration du logiciel et du matériel.

NOTE 2 Les essais boîte noire visent à vérifier le comportement dynamique en conditions de fonctionnement réel, et à révéler les défaillances pour satisfaire à la spécification fonctionnelle, et évaluer l'utilité et la robustesse. Les essais boîte grise s'apparentent aux essais boîte noire, mais surveillent en outre le ou les paramètres d'essai pertinents à l'intérieur du module logiciel.

8.3.6 Essai du logiciel – Niveau logiciel 1

8.3.6.1 Niveau logiciel 1 – Généralités

Les essais du logiciel ont pour principal objet de vérifier que les fonctionnalités détaillées dans la spécification de conception du logiciel sont assurées.

Le principal résultat des essais du logiciel est un document (un rapport d'essai, par exemple) contenant des cas d'essai et des résultats d'essai permettant d'évaluer la couverture de l'essai.

Les essais du logiciel doivent également inclure une simulation de défaillance et la réaction associée en fonction de l'intégrité de sécurité exigée.

Si des cartes d'entrée types ou des modules logiciels qui intègrent une détection des défaillances et sa réaction sont utilisés (divergence des signaux d'entrée ou contact de sortie en retour, par exemple), l'essai de cette détection des défaillances et de sa réaction n'est pas nécessaire. Dans ce cas, seule l'intégration des cartes d'entrée ou des modules logiciels conformément à la spécification du fabricant doit être vérifiée par essai.

Les essais du logiciel peuvent être réalisés dans le cadre de la validation du système si les essais sont réalisés sur le matériel cible.

Les essais fonctionnels doivent être appliqués comme mesure de base. Il convient de soumettre le code à l'essai par simulation, dans la mesure du possible.

Il est recommandé de définir les lignes directrices ou procédures générales des essais du logiciel relatif à la sécurité. Il convient que ces lignes directrices ou procédures contiennent:

- les types d'essais à réaliser;
- la spécification du matériel d'essai, y compris les outils, le logiciel support et la description de la configuration;
- la gestion du versionnage de logiciel pendant les essais et la correction du logiciel relatif à la sécurité;
- les actions correctives sur les essais non concluants;
- les critères de réalisation de l'essai par rapport aux fonctions ou exigences associées; les emplacements physiques de l'essai, comme la simulation informatique, sur une table ou en laboratoire, en usine ou sur la machine.

8.3.6.2 Programme et exécution des essais – Niveau logiciel 1

Le programme d'essai reposant sur des cas d'essai doit inclure ce qui suit:

- définition des rôles et des responsabilités par nom;
- essais d'installation;
- essais fonctionnels.

8.3.7 Documentation – Niveau logiciel 1

Toutes les activités de cycle de vie doivent être traçables en amont et en aval de la spécification de la ou des fonctions de sécurité et tout au long de la réalisation du plan de validation.

Les entrées et sorties de toutes les phases du cycle de vie de sécurité du logiciel doivent être documentées et mises à disposition des personnes compétentes.

Les résultats des activités d'essai et les actions correctives réalisées doivent être documentés.

8.3.8 Processus de gestion de la configuration et des modifications – Niveau logiciel 1

Toute modification ou correction du logiciel doit être soumise à une analyse d'impact qui identifie toutes les parties affectées du logiciel et les activités nécessaires de conception, d'examen et d'essai à exécuter de nouveau pour confirmer que les exigences de sécurité du logiciel concernées sont toujours satisfaites.

Les processus de gestion de configuration et des modifications doivent être définis et documentés. Cette procédure doit comprendre au moins les points suivants:

- les articles gérés par la configuration, c'est-à-dire au moins: les exigences de sécurité du logiciel, la conception préliminaire et détaillée du logiciel, les modules de code source, les plans, les procédures et les résultats des essais de validation;
- les règles d'identification qui identifient sans équivoque chaque module logiciel ou élément de configuration;
- les processus de modification compréhensibles par demande à travers la mise en œuvre.

Pour chaque article de la configuration, il doit être possible d'identifier toutes les corrections qui peuvent avoir eu lieu et les versions de tous les éléments associés.

NOTE 1 Il s'agit d'être capable de conserver l'historique du développement de chaque article: les modifications réalisées, les raisons et les dates de ces modifications.

La gestion de configuration logicielle doit permettre l'obtention d'une identification d'une version logicielle, précise et unique. Il convient que la gestion de configuration s'associe à tous les articles (et leur version) nécessaires à la démonstration de la sécurité fonctionnelle.

Tous les articles dans la configuration logicielle doivent être couverts par la procédure de gestion de configuration avant d'être soumis aux essais ou lorsque l'analyste demande une dernière évaluation d'une dernière version.

NOTE 2 L'objectif est ici d'assurer que la procédure d'évaluation est réalisée sur un logiciel dont tous les éléments sont dans un état précis. Toute correction ultérieure peut nécessiter une révision du logiciel telle qu'elle puisse être identifiable par l'analyste.

Les procédures d'archivage du logiciel et de ses données associées (méthodes pour fichiers de sauvegarde et archives) doivent être établies.

NOTE 3 Ces sauvegardes et archives peuvent être utilisées pour actualiser et modifier le logiciel pendant sa durée de vie d'exploitation.

8.4 Niveau logiciel 2

8.4.1 Cycle de vie de sécurité du logiciel – Niveau logiciel 2

8.4.1.1 Niveau SIL maximal atteignable – Niveau logiciel 2

Le niveau SIL maximal atteignable pour le niveau logiciel 2 est SIL 2.

8.4.1.2 Modèle de cycle de vie de sécurité du logiciel – Niveau logiciel 2

Un modèle de cycle de vie de sécurité du logiciel résolu en différentes phases doit être utilisé (modèle en V, par exemple), y compris les activités de gestion et de documentation permettant d'obtenir le niveau de sécurité exigé.

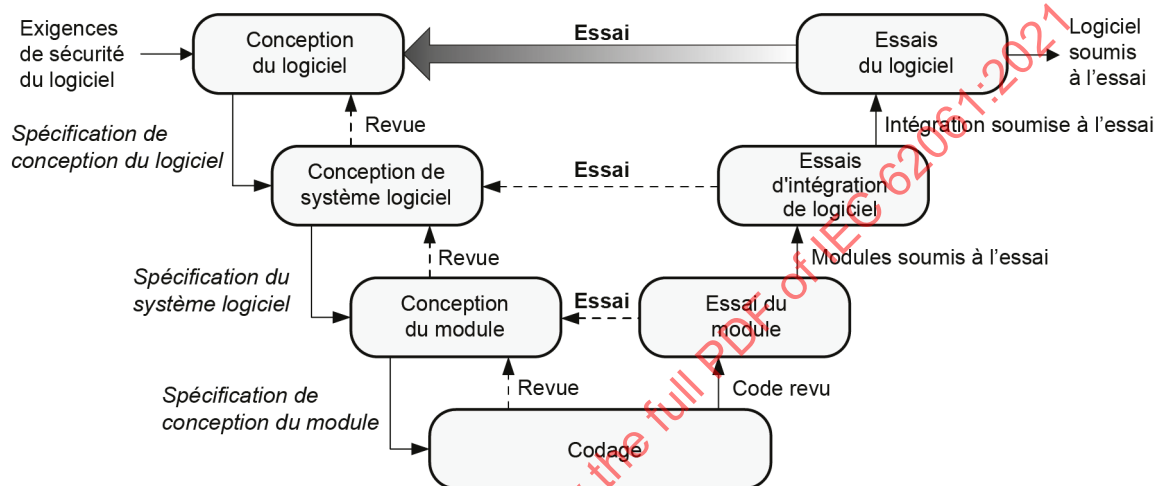
Un modèle de cycle de vie du logiciel peut être utilisé à condition que tous les objectifs et toutes les exigences du présent 8.4 soient satisfaits. Le logiciel relatif à la sécurité doit être validé comme cela est indiqué en 9.5.3.

Le niveau logiciel 2 présente une complexité plus importante comparée à celle du niveau logiciel 1 en raison de l'utilisation de langages de programmation totalement variables. Par conséquent, le modèle en V plus détaillé de la Figure 14 s'applique.

NOTE 1 Le modèle en V est un modèle statique utilisé pour structurer la conception du logiciel en petites parties. Il n'introduit aucune séquence de création de spécifications ni de mise en œuvre. Le côté gauche représente les exigences, c'est-à-dire les choses à atteindre. Le côté droit détaille les essais du logiciel.

NOTE 2 Sur le côté gauche du modèle en V, chaque phase est examinée. Il s'agit de vérifier la sortie d'une phase dans le modèle en V par rapport aux exigences de l'entrée de la même phase. La flèche "Examiner" représente la première étape de la vérification du logiciel. De plus amples informations sur le niveau d'indépendance de l'examen et de l'essai ou de la vérification sont disponibles dans l'Annex J.

NOTE 3 Le choix des techniques et processus de gestion de projet peut s'effectuer de sorte que ceux-ci soient appropriés pour la taille et le domaine d'application du projet.



IEC

Figure 14 – Modèle en V du cycle de vie de sécurité du logiciel pour le niveau logiciel 2

NOTE 4 Dans les modèles en V, la flèche "Essai" représente les résultats des cas d'essai en fonction de la spécification et, de plus, la nécessité d'exigences et de spécifications plus précises concernant le cas d'essai.

8.4.1.3 Utilisation des outils – Niveau logiciel 2

Un ensemble adapté d'outils doit être choisi (gestion de configuration, simulation et matériel d'essai avec générateur d'essai, par exemple). De préférence, il convient d'utiliser les outils recommandés par le fabricant. La disponibilité des outils adaptés pour le service, la mise à jour de la machine et la paramétrisation sur la durée de vie du système de commande relatif à la sécurité doivent être prises en considération. Soit les outils fournis par le fabricant du matériel sont utilisés, soit la pertinence des outils doit être expliquée et documentée.

La pertinence doit être démontrée comme suit:

- une analyse réalisée pour identifier les possibles effets d'une défaillance provoquée par ces outils dans la chaîne d'outils; et
- des mesures appropriées permettant d'éviter et de maîtriser les anomalies, à choisir et à appliquer, et dont l'efficacité est vérifiée par des essais rigoureux, les résultats étant documentés.

NOTE 1 Le caractère approprié des mesures permettant d'éviter et de maîtriser les anomalies dépend de la sévérité des conséquences d'une défaillance. Cette évaluation repose sur une analyse. Pour procéder à cette analyse, il est nécessaire d'avoir les connaissances relatives à l'application de l'outil support et à la machine.

NOTE 2 Les effets des défaillances peuvent varier en fonction des outils support. Par conséquent, l'IEC 61508-4 distingue trois catégories d'outils support hors ligne utilisés dans le cycle de vie du développement du logiciel. Cette distinction peut faire partie de l'analyse.

NOTE 3 Voir l'IEC 61508-4 pour une définition des outils support et des exemples.

NOTE 4 Le présent document ne spécifie aucune mesure permettant d'éviter ou de maîtriser les anomalies des outils support hors lignes. Pour des exemples, voir l'IEC 61508-3:2010, 7.4.4.

8.4.2 Conception du logiciel – Niveau logiciel 2

8.4.2.1 Niveau logiciel 2 – Généralités

La spécification de conception du logiciel doit être développée sur la base des exigences de sécurité du logiciel, et gérée tout au long du cycle de vie du SCS.

8.4.2.2 Exigences de sécurité du logiciel – Niveau logiciel 2

Pour prendre en charge le processus de conception du logiciel, les informations suivantes doivent être prises en considération:

- a) spécification de la ou des fonctions de sécurité (voir 5.2);
- b) configuration ou architecture du SCS (architecture du matériel, schéma de câblage, entrées et sorties relatives à la sécurité, par exemple);
- c) exigences de temps de réponse;
- d) interfaces et commandes de l'opérateur (interrupteurs, joysticks, sélecteur de mode, cadrans, dispositifs de commande tactiles, claviers, etc.);
- e) modes de fonctionnement adaptés de la machine;
- f) exigences relatives aux diagnostics du matériel, y compris les caractéristiques des capteurs, des actionneurs finaux, etc.;
- g) effets des tolérances mécaniques (des capteurs et/ou de leurs contreparties sensibles, par exemple);
- h) lignes directrices en matière de codage.

Lors de l'application du niveau logiciel 2, les tableaux de l'IEC 61508-3:2010, Annexe A et Annexe B doivent être pris en considération s'il est approprié d'utiliser d'autres techniques et mesures présentant une efficacité équivalente. L'IEC 61508-7 donne des informations supplémentaires.

La conception et le choix du langage choisi pour satisfaire au niveau SIL exigé du SCS doivent être appropriés pour l'application.

La conception doit inclure l'autosurveillance du flux de commande et du flux de données correspondant au niveau SIL du SCS. En cas de détection de défaillance, des actions appropriées doivent être accomplies pour réaliser ou maintenir un état de sécurité.

8.4.2.3 Spécification de conception du logiciel – Niveau logiciel 2

La spécification de conception du logiciel doit être déduite des exigences de sécurité du logiciel du SCS.

La spécification de conception du logiciel doit être:

- structurée, examinable, testable, compréhensible, actualisable et manœuvrable;
- développée pour chaque sous-système sur la base de la spécification et de l'architecture du SCS;
- suffisamment détaillée pour permettre la conception et la mise en œuvre du SCS afin d'atteindre le niveau de sécurité (SIL) exigé, ainsi que la vérification et les essais.

- traçable par rapport à la spécification des exigences de sécurité du logiciel du SCS. Cela signifie que la spécification est par nature compréhensible de sorte qu'une autre personne (n'étant pas spécialiste en matière de logiciels, par exemple) puisse vérifier si elle correspond aux exigences de sécurité du logiciel des fonctions de sécurité définies dans l'appréciation du risque.
- exempte de toute terminologie ambiguë et descriptions non pertinentes.

Il doit être possible d'associer directement les entrées de la spécification de conception du logiciel aux sorties souhaitées, et inversement. Le cas échéant, des méthodes semi-formelles aisément lisibles (des tables de cause à effet, des tableaux ou schémas logiques, des blocs fonctionnels ou des diagrammes de séquence, par exemple) doivent être utilisées dans la documentation.

NOTE 1 Cela dépend, le cas échéant, du nombre de fonctions de sécurité impliquées dans le programme. À chaque fois que le nombre total de fonctions à l'intérieur du programme dépasse 3, il est considéré comme étant approprié.

Les éléments suivants doivent être précisés dans la spécification de conception du logiciel:

- a) la logique des fonctions de sécurité, y compris les entrées et sorties relatives à la sécurité et les diagnostics corrects des anomalies détectées. Les méthodes possibles incluent, entre autres, la table de cause à effet, une description écrite ou des blocs fonctionnels;

NOTE 2 Les anomalies peuvent également être détectées par le matériel (incohérence de signal détectée par la carte d'entrée, par exemple).

- b) les cas d'essai, qui incluent:

- la ou les valeurs d'entrée particulières pour lesquelles l'essai est réalisé et les résultats d'essai prévus, y compris les critères de réussite/d'échec;
- l'insertion ou l'injection ou les injections d'anomalie.

NOTE 3 Pour les fonctions simples, le ou les cas d'essai peuvent être donnés de manière implicite par la spécification de la fonction de sécurité.

- c) les fonctions de diagnostic des dispositifs d'entrée (les éléments de détection et les interrupteurs, par exemple) et les éléments de commande finale (les solénoïdes, les relais ou les contacteurs, par exemple);
- d) les fonctions qui permettent à la machine d'atteindre ou de maintenir un état de sécurité;
- e) les fonctions liées à la détection, la signalisation et le traitement des anomalies;
- f) les fonctions liées aux essais périodiques du ou des SCS en ligne et hors ligne;
- g) les fonctions qui empêchent une modification non autorisée du SCS (un mot de passe, par exemple);
- h) les interfaces vers des éléments non SCS;
- i) le temps de réponse de la fonction de sécurité.

NOTE 4 Des recommandations relatives à la documentation du logiciel sont données dans l'IEC 61508 et l'ISO/IEC/IEEE 26512.

Dans la mesure du possible, il est recommandé d'utiliser des modules logiciels types dans les limites de la spécification de conception du logiciel.

Dans le cas des sous-fonctions de sécurité types (IEC 61800-5-2, par exemple), il est recommandé de faire référence à la spécification fournie par le fabricant.

Les informations présentées dans la spécification de conception du logiciel doivent être examinées et, le cas échéant, révisées de manière à assurer que les exigences de sécurité du logiciel (voir 8.4.2.2) sont correctement spécifiées.

8.4.3 Conception du système logiciel – Niveau logiciel 2

8.4.3.1 Niveau logiciel 2 – Généralités

La conception du système logiciel commence par la définition de l'architecture. L'architecture du logiciel doit être établie conformément à la spécification de conception du logiciel. L'architecture du logiciel définit les éléments et sous-systèmes essentiels du logiciel, la manière dont ils sont interconnectés et dont les attributs exigés sont obtenus. Elle définit également le comportement général du logiciel et la manière dont les éléments logiciels interfacent et interagissent. Des exemples d'éléments logiciels essentiels sont les systèmes d'exploitation, les bases de données, les sous-systèmes d'entrée/sortie, les sous-systèmes de communication, les programmes d'application, les outils de programmation et de diagnostic, etc.

La conception du système logiciel doit suivre une approche modulaire avec une taille de module logiciel limitée, une interface totalement définie et un point d'entrée/un point de sortie dans les sous-routines et les fonctions. La fonction ou l'objet de chaque module doit être simplement et clairement défini. La taille maximale du module doit être limitée à une fonction de sécurité complète.

Les techniques de programmation suivantes doivent être utilisées pour éviter les défaillances systématiques:

- contrôle des limites et contrôle de vraisemblance des variables et des paramètres de configuration;
- contrôle logique ou temporel de la séquence de programme pour détecter une séquence de programme défectueuse: Une séquence de programme est défectueuse si chaque élément d'un programme (modules logiciels, sous-programmes ou commandes, par exemple) est traité dans la mauvaise séquence ou période ou si l'horloge du processeur est erronée (voir l'IEC 61508-7:2010, Article A.9);
- limitation du nombre ou de l'étendue des variables globales.

NOTE Pour le niveau logiciel 2, voir l'Annexe G de l'IEC 61508-7:2010 pour des recommandations relatives à l'architecture et la conception orientées objet.

8.4.3.2 Spécification de conception du système logiciel – Niveau logiciel 2

Une spécification de conception du système logiciel doit être fournie en résultat de la conception du système logiciel. Elle doit expliquer les principaux aspects du logiciel, comme ceux indiqués dans la liste suivante, par exemple:

- l'architecture du logiciel qui définit la structure décidée en vue de satisfaire à la spécification de conception du logiciel;
- les données globales;
- les bibliothèques de données utilisées;
- les modules logiciels préexistants utilisés;
- les fonctions de diagnostic (internes, externes);
- les outils de programmation, y compris les informations qui identifient l'outil de manière unique;
- les cas d'essai et procédures d'intégration, y compris la spécification de l'environnement d'essai, le logiciel support, la description de la configuration et les procédures d'actions correctives en cas de défaillance d'un essai.

Les informations contenues dans la spécification du système logiciel doivent être examinées par rapport à la spécification de conception du logiciel.

8.4.4 Conception du module – Niveau logiciel 2

8.4.4.1 Niveau logiciel 2 – Généralités

Lorsque des modules de bibliothèque logicielle développés précédemment doivent être utilisés pour la conception, leur capacité à satisfaire à la spécification des exigences de sécurité du logiciel doit être démontrée. Les contraintes imposées par l'environnement de développement du logiciel précédent (par exemple les dépendances au système d'exploitation et au compilateur) doivent être évaluées.

8.4.4.2 Informations d'entrée – Niveau logiciel 2

Pour les modules logiciels, les informations suivantes doivent être disponibles dans la spécification de conception du système logiciel:

- a) la description du module;
- b) l'interface du module (entrées et sorties avec les types de données et, le cas échéant, avec les plages de données);
- c) les bibliothèques de modules utilisées;
- d) les règles de codage spéciales.

8.4.4.3 Spécification de conception du module – Niveau logiciel 2

La spécification de conception du module doit contenir les informations suivantes:

- a) la description de la logique (c'est-à-dire la fonctionnalité) de chaque module;
- b) les interfaces d'entrée et de sortie intégralement définies pour chaque module;
- c) les formats et les plages de valeurs des données d'entrée et de sortie et leur relation avec les modules;
- d) les cas d'essai qui doivent inclure le fonctionnement normal et normal en extérieur;

NOTE Même si les cas d'essai comprennent souvent les essais individuels des paramètres dans leurs plages spécifiées, une variation de combinaison de ces paramètres peut introduire un fonctionnement intempestif.

- e) documentation des interruptions.

Ces informations doivent être examinées par rapport aux informations d'entrée (voir 8.4.4.2).

8.4.5 Codage – Niveau logiciel 2

Le logiciel doit être développé conformément aux spécifications de conception et aux règles de codage. Les règles de codage peuvent être des normes de l'industrie reconnues ou peuvent être internes au fabricant. Le code doit être examiné par rapport aux spécifications de conception et aux règles de codage.

NOTE 1 Les règles de codage sont destinées à limiter la liberté de programmation, afin d'éviter que le code de programme ne devienne incompréhensible et de limiter la probabilité d'entrée du programme dans des états imprévus.

NOTE 2 Généralement, les règles de codage définissent un sous-ensemble d'un langage de programmation ou utilisent un langage de programmation fortement typé (voir l'IEC 61508-7:2010, C.4.1).

La sortie du codage doit être composée

- de la liste du code source (échelle, blocs fonctionnels, modèles, par exemple);
- du rapport de revue de code.

Certaines règles de codage classiques à appliquer sont, entre autres, les suivantes:

- La structure du programme est aussi facile et claire que possible.
- Il convient que la structure du programme soit telle que le flux logique commence en haut et suive la séquence effective.
- Il convient que chaque partie comporte des commentaires suffisants de manière prédéfinie.
- Il convient d'utiliser les mêmes noms pour les paramètres que ceux utilisés lors de la conception.
- Il convient que le nom représente de manière claire la fonction du paramètre.
- Il convient de disposer d'un état prédéfini.
- Il convient de limiter l'utilisation des termes définir/redéfinir pour les fonctions de sécurité.
- Il convient que les sorties de sécurité ne soient attribuées qu'une seule fois à l'intérieur d'un programme.
- Les paramètres non relatifs à la sécurité ne doivent pas être utilisés pour éviter des fonctions de sécurité.

8.4.6 Essai du module – Niveau logiciel 2

Chaque module qui n'a pas été préalablement évalué doit être soumis à l'essai en fonction des cas d'essai définis dans la conception du module par des essais fonctionnels et des essais boîte noire, boîte grise ou boîte blanche, selon le cas.

Si le module ne satisfait aux essais, une action corrective prédéfinie doit être réalisée.

Les résultats d'essai et les actions correctives doivent être documentés.

NOTE 1 Les essais fonctionnels visent à révéler les défaillances pendant les phases de spécification et de conception, et à éviter les défaillances lors de la mise en œuvre et de l'intégration du logiciel et du matériel.

NOTE 2 Les essais boîte noire visent à vérifier le comportement dynamique en conditions de fonctionnement réel, et à révéler les défaillances pour satisfaire à la spécification fonctionnelle, et évaluer l'utilité et la robustesse. Les essais boîte grise s'apparentent aux essais boîte noire, mais surveillent en outre le ou les paramètres d'essai pertinents à l'intérieur du module logiciel.

Les essais de module doivent au moins s'appuyer sur une analyse et des essais dynamiques.

8.4.7 Essai d'intégration du logiciel – Niveau logiciel 2

Le logiciel doit être soumis à l'essai en fonction des cas d'essai d'intégration. Les résultats des essais d'intégration du logiciel doivent être documentés.

NOTE Ces essais ont pour objet de démontrer que tous les modules logiciels et éléments/sous-systèmes logiciels interagissent correctement pour réaliser leur fonction prévue et ne pas réaliser de fonctions imprévues. Cela n'implique pas de soumettre à l'essai toutes les combinaisons d'entrée ni toutes les combinaisons de sortie. Il peut être suffisant de soumettre à l'essai toutes les classes d'équivalence ou de procéder à des essais en fonction de la structure. Une analyse aux valeurs limites ou une analyse du flux de commande peut réduire les cas d'essai à un nombre acceptable. Des programmes analysables rendent les exigences plus faciles à satisfaire.

8.4.8 Essai du logiciel – Niveau logiciel 2

8.4.8.1 Niveau logiciel 2 – Généralités

Les essais du logiciel ont pour principal objet de vérifier que les fonctionnalités détaillées dans la spécification de conception du logiciel sont assurées.

NOTE Cela peut impliquer de soumettre à l'essai toutes les combinaisons d'entrée et/ou toutes les combinaisons de sortie.

Le principal résultat des essais du logiciel est un document (un rapport d'essai, par exemple) contenant des cas d'essai et des résultats d'essai permettant d'évaluer la couverture de l'essai.

Les essais du logiciel doivent également inclure une simulation de défaillance et la réaction associée en fonction de l'intégrité de sécurité exigée.

Si des cartes d'entrée types ou des modules logiciels qui intègrent une détection des défaillances et sa réaction sont utilisés (divergence des signaux d'entrée ou contact de sortie en retour, par exemple), l'essai de cette détection des défaillances et de sa réaction n'est pas nécessaire. Dans ce cas, seule l'intégration des cartes d'entrée ou des modules logiciels conformément à la spécification du fabricant doit être vérifiée par essai.

Les essais du logiciel peuvent être réalisés dans le cadre de la validation du système si les essais sont réalisés sur le matériel cible.

Les essais fonctionnels doivent être appliqués comme mesure de base. Il convient de soumettre le code à l'essai par simulation, dans la mesure du possible.

Il est recommandé de définir les lignes directrices ou procédures générales des essais du logiciel relatif à la sécurité. Il convient que ces lignes directrices ou procédures contiennent:

- les types d'essais à réaliser;
- la spécification du matériel d'essai, y compris les outils, le logiciel support et la description de la configuration;
- la gestion du versionnage de logiciel pendant les essais et la correction du logiciel relatif à la sécurité;
- les actions correctives sur les essais non concluants;
- les critères de réalisation de l'essai par rapport aux fonctions ou exigences associées; les emplacements physiques de l'essai, comme la simulation informatique, sur une table ou en laboratoire, en usine ou sur la machine.

8.4.8.2 Programme et exécution des essais – Niveau logiciel 2

Le programme d'essai reposant sur des cas d'essai doit inclure ce qui suit:

- définition des rôles et des responsabilités par nom;
- essais d'installation;
- essais fonctionnels.

Les essais de logiciel incluent deux types d'activités:

- Une analyse statique: analyse de la documentation du logiciel (revue, examen, lecture croisée, analyse du flux de commande ou analyse du flux de données, par exemple).
- Essais dynamiques: exécution du logiciel de manière maîtrisée et systématique, de façon à démontrer la présence du comportement exigé et l'absence de comportement indésirable. Cela comprend, en particulier, les essais fonctionnels, les essais boîte noire et les essais boîte grise.

Dans les premières phases du cycle de vie du logiciel, la vérification est statique. Les essais dynamiques deviennent possibles lorsque le code est généré. Pour vérifier le résultat des activités de cycle de vie du logiciel, la combinaison des deux activités est exigée. Pour une description plus détaillée de l'analyse statique et des essais dynamiques, voir l'IEC 61508-3.

Les éléments suivants sont exigés pour la vérification et les essais du logiciel relatif à la sécurité:

- l'analyse statique doit être réalisée et documentée dans tous les cas;
- les essais dynamiques doivent être réalisés et documentés;

- si le logiciel est exigé pour une fonction de sécurité jusqu'au niveau SIL 1 et qu'il ne fait pas l'objet d'un essai dynamique, cela doit être justifié en fonction de la simplicité structurelle du logiciel;
- pour les essais dynamiques, chaque sous-programme (sous-routine ou fonction) doit avoir été appelé au moins une fois (points d'entrée) pendant l'essai;
- pour le logiciel, qui est exigé pour une fonction de sécurité de niveau SIL 2, toutes les déclarations doivent être exécutées au moins une fois pendant l'essai dynamique;
- si un logiciel est utilisé dans des fonctions de diagnostic pour la maîtrise des défaillances aléatoires du matériel, les essais dynamiques doivent concerner la mise en œuvre correcte des diagnostics (par un essai d'insertion d'anomalies, par exemple);
- les essais dynamiques doivent inclure un essai final réalisé sur le matériel cible.

8.4.9 Documentation – Niveau logiciel 2

Toutes les activités de cycle de vie doivent être traçables en amont et en aval de la spécification de la ou des fonctions de sécurité et tout au long de la réalisation du plan de validation.

Les entrées et sorties de toutes les phases du cycle de vie de sécurité du logiciel doivent être documentées et mises à disposition des personnes compétentes.

Les résultats des activités d'essai et les actions correctives réalisées doivent être documentés.

8.4.10 Processus de gestion de la configuration et des modifications – Niveau logiciel 2

Toute modification ou correction du logiciel doit être soumise à une analyse d'impact qui identifie toutes les parties affectées du logiciel et les activités nécessaires de conception, d'examen et d'essai à exécuter de nouveau pour confirmer que les exigences de sécurité du logiciel concernées sont toujours satisfaites.

Les processus de gestion de configuration et des modifications doivent être définis et documentés. Cette procédure doit comprendre au moins les points suivants:

- les articles gérés par la configuration, c'est-à-dire au moins: les exigences de sécurité du logiciel, la conception préliminaire et détaillée du logiciel, les modules de code source, les plans, les procédures et les résultats des essais de validation;
- les règles d'identification qui identifient sans équivoque chaque module logiciel ou élément de configuration;
- les processus de modification compréhensibles par demande à travers la mise en œuvre.

Pour chaque article de la configuration, il doit être possible d'identifier toutes les corrections qui peuvent avoir eu lieu et les versions de tous les éléments associés.

NOTE 1 Il s'agit d'être capable de conserver l'historique du développement de chaque article: les modifications réalisées, les raisons et les dates de ces modifications.

La gestion de configuration logicielle doit permettre l'obtention d'une identification d'une version logicielle, précise et unique. La gestion de configuration doit s'associer à tous les articles (et leur version) nécessaires à la démonstration de la sécurité fonctionnelle.

Tous les articles dans la configuration logicielle doivent être couverts par la procédure de gestion de configuration avant d'être soumis aux essais ou lorsque l'analyste demande une dernière évaluation d'une dernière version.

NOTE 2 L'objectif ici est d'assurer que la procédure d'évaluation est réalisée sur un logiciel dont tous les éléments sont dans un état précis. Toute correction ultérieure peut nécessiter une révision du logiciel telle qu'elle puisse être identifiable par l'analyste.

Les procédures d'archivage du logiciel et de ses données associées (méthodes pour fichiers de sauvegarde et archives) doivent être établies.

NOTE 3 Ces sauvegardes et archives peuvent être utilisées pour actualiser et modifier le logiciel pendant sa durée de vie d'exploitation.

9 Validation

9.1 Principes de validation

Dans le présent document, la validation a pour objet de confirmer que le SCS respecte la spécification des exigences de sécurité donnée à l'Article 5 et les informations d'utilisation données en 10.3.

NOTE 1 Dans le présent document, la validation est limitée au SCS conçu ou à l'une de ses parties prenant en charge les fonctions de sécurité exigées par la stratégie de réduction de risque au niveau de la machine (voir l'ISO 12100). Le résultat de la validation du SCS est destiné à faire partie de la validation globale de la machine.

NOTE 2 Dans certains cas, la validation de la sécurité peut être seulement réalisée après l'installation définitive (par exemple, lorsque le développement du logiciel d'application n'est pas finalisé).

Les activités de validation consistent à rassembler et vérifier la disponibilité de la preuve en démontrant l'exhaustivité de chaque activité de conception identifiée dans le plan de sécurité.

La validation à appliquer au SCS comprend l'examen (par analyse, par exemple) et les essais du SCS afin de vérifier qu'il satisfait aux exigences formulées dans la spécification des exigences de sécurité (selon l'Article 5).

La validation doit démontrer que le SCS satisfait aux exigences et, en particulier, à ce qui suit:

- a) les exigences fonctionnelles spécifiées des fonctions de sécurité fournies par cette partie (voir 5.2), telles que définies par la justification de conception;
- b) les exigences du niveau SIL spécifié.

La validation doit être réalisée par des personnes indépendantes de la conception du SCS.

NOTE 3 "Personne indépendante" ne veut pas nécessairement dire qu'un essai tiers est exigé.

Il convient de commencer l'analyse dès que possible dans le processus de conception et parallèlement à celui-ci.

NOTE 4 Les problèmes peuvent ensuite être corrigés de manière précoce pendant qu'il est relativement aisé de le faire, c'est-à-dire au cours de la "conception et de la réalisation technique de la fonction de sécurité" et de "l'évaluation du niveau SIL". Il peut s'avérer nécessaire de différer certaines parties de l'analyse tant que la conception n'est pas bien développée.

La Figure 15 donne un aperçu du processus de validation: la validation consiste à procéder à une analyse (voir 9.2) et à des essais fonctionnels (voir 9.3) dans des conditions prévisibles conformément au plan de validation. L'équilibre entre l'analyse et les essais doit être justifié. Pour les architectures avec fonction de diagnostic, la validation de la fonction de sécurité doit également inclure les essais dans les conditions d'anomalie pour démontrer que la réaction à l'anomalie est initiée par la fonction de diagnostic mise en œuvre.

Selon le cas, en raison de la taille du système, de sa complexité ou des effets de son intégration au système de commande (des machines), il convient de prendre des dispositions particulières pour

- valider le sous-système séparément avant l'intégration, y compris la simulation des signaux d'entrée et de sortie appropriés, et
- valider les effets de l'intégration des parties relatives à la sécurité dans le reste du système de commande dans le cadre de son utilisation dans la machine.

Dans la Figure 15, "Modification de la conception" fait référence au processus de conception. Si la validation ne peut pas aboutir, des modifications de la conception sont nécessaires. Le cas échéant, il convient alors de répéter la validation du SCS. Il convient de répéter ce processus jusqu'à ce que chaque fonction de sécurité du SCS soit correctement validée.

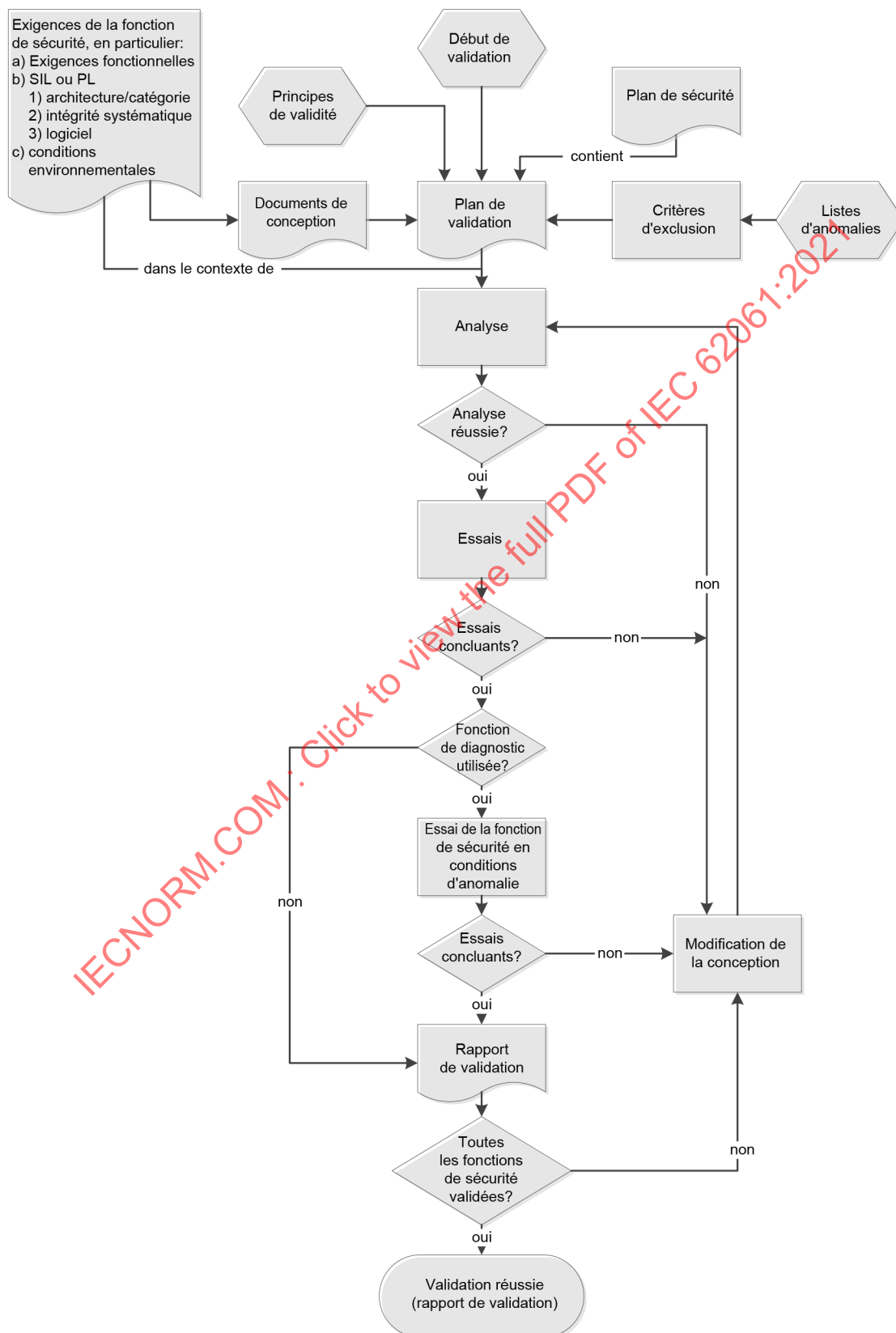


Figure 15 – Aperçu du processus de validation

9.1.1 Plan de validation

Le plan de validation doit identifier et décrire les exigences de réalisation du processus de validation. Le plan de validation doit également identifier les moyens à utiliser pour valider les fonctions de sécurité spécifiées. Il doit établir, le cas échéant:

- a) l'identité des documents de spécification,
- b) les conditions opérationnelles et environnementales pendant les essais,
- c) les analyses et essais à appliquer,
- d) la référence aux normes d'essai à appliquer,
- e) les personnes ou parties chargées de chaque étape du processus de validation, et
- f) le matériel exigé.

Pour les sous-systèmes déjà validés selon la même spécification, il est nécessaire de faire uniquement référence à cette précédente validation.

NOTE Des informations sur le niveau d'indépendance de la validation figurent à l'Annex J.

9.1.2 Utilisation des listes d'anomalies génériques

La validation implique de prendre en considération le comportement du SCS pour toutes les anomalies à considérer. La base de la prise en considération des anomalies est donnée dans les tableaux des listes d'anomalies de l'ISO 13849-2:2012, Annexe A à Annexe D, qui reposent sur l'expérience et qui contiennent:

- les composants/éléments à inclure (conducteurs/câbles, par exemple),
- les anomalies à prendre en compte (les courts-circuits entre les conducteurs, par exemple),
- les exclusions d'anomalie admises, en tenant compte des aspects liés à l'environnement, l'exploitation et l'application, et
- une section de remarques donnant les raisons justifiant les exclusions d'anomalie.

Seules les anomalies permanentes sont prises en compte dans les listes d'anomalies.

9.1.3 Listes d'anomalies spécifiques

Si nécessaire, une liste spécifique d'anomalies relatives au produit doit être générée sous la forme d'un document de référence pour la validation du ou des sous-systèmes et/ou éléments de sous-système.

NOTE La liste peut reposer sur une ou plusieurs listes génériques appropriées présentées dans les Annexes A à D de l'ISO 13849-2:2012.

Si la liste spécifique d'anomalies relatives au produit repose sur la ou les listes génériques, elle doit indiquer

- i) les anomalies issues de la ou des listes génériques à inclure,
- j) d'autres anomalies pertinentes à inclure, mais qui ne sont pas indiquées dans la liste générique (les défaillances de cause commune, par exemple),
- k) les anomalies issues de la ou des listes génériques qui peuvent être exclues au motif que les critères indiqués dans la ou les listes génériques sont satisfaits (voir 7.3.3),
- l) et, exceptionnellement, toutes les autres anomalies dont la ou les listes génériques ne permettent pas l'exclusion, mais dont la justification d'une exclusion est présentée (voir 7.3.3).

Si cette liste ne repose pas sur une ou plusieurs listes génériques, le concepteur doit justifier les exclusions d'anomalie.

9.1.4 Informations pour la validation

Les informations exigées pour la validation varient en fonction de la technologie utilisée, des contraintes architecturales et du niveau SIL à démontrer, de la justification de conception du système et de la contribution du SCS à la réduction du risque. Les documents contenant des informations suffisantes issues de la liste suivante doivent être inclus, selon le cas, dans la validation afin de démontrer que les parties relatives à la sécurité réalisent les fonctions de sécurité spécifiées selon le niveau SIL et les contraintes architecturales exigés:

- a) spécification des caractéristiques exigées de chaque fonction de sécurité, en particulier le niveau SIL et les contraintes architecturales exigés;
- b) les schémas et spécifications (des parties mécaniques, hydrauliques et pneumatiques, des cartes de circuit imprimé, des cartes assemblées, du câblage interne, de l'enveloppe, des matériaux, du montage, par exemple);
- c) le ou les schémas fonctionnels avec une description des blocs;
- d) le ou les schémas des circuits, y compris les interfaces/connexions;
- e) la description fonctionnelle du ou des schémas des circuits;
- f) le ou les diagrammes de séquence temporelle pour les composants de commutation, les signaux relatifs à la sécurité;
- g) la description des caractéristiques correspondantes des composants déjà validés;
- h) pour les parties relatives à la sécurité autres que celles indiquées en g), les listes de composants avec les désignations d'éléments, les valeurs assignées, les tolérances, les contraintes de fonctionnement pertinentes, la désignation du type, les données relatives au taux de défaillance et le fabricant du composant, ainsi que toutes les autres données pertinentes pour la sécurité;
- i) l'analyse de toutes les anomalies pertinentes selon 9.1.2 et 9.1.3, comme celles figurant dans les tableaux de l'ISO 13849-2:2012, Annexe A à Annexe D, y compris la justification de toutes les anomalies exclues;
- j) une analyse de l'influence des matériaux traités;
- k) les informations d'utilisation (manuel d'installation et d'exploitation/manuel d'instructions, par exemple).

Si le logiciel concerne la ou les fonctions de sécurité, la documentation du logiciel doit inclure

- une spécification claire et sans équivoque, qui établit l'intégrité de sécurité que le logiciel doit atteindre,
- la preuve que le logiciel est conçu pour atteindre le niveau SIL exigé (voir 9.5.4), et
- les détails de la vérification (en particulier les rapports d'essai) réalisée pour démontrer que le niveau SIL exigé est atteint.

Les informations sont exigées quant à la manière de déterminer le niveau SIL et la *PFH*. La documentation des aspects quantifiables doit inclure

- l'architecture de sous-système simple selon 7.5.2,
- les paramètres de fiabilité de la détermination ($MTTF_D$ ou λ_D des éléments de sous-système et la CCF), et
- la détermination des contraintes architecturales.

Des informations sont exigées pour la documentation sur les aspects systématiques du SCS. Des informations sont exigées pour décrire la manière dont la combinaison de plusieurs sous-systèmes permet d'atteindre un niveau SIL conforme à celui qui est exigé.

9.1.5 Consignation de la validation

La validation par analyse et essai doit être consignée (voir également l'Article 10). La documentation appropriée doit indiquer:

- la version du plan de validation utilisée et la version de la fonction de sécurité soumise à l'essai;
- la fonction de sécurité en essai (ou en cours d'analyse), ainsi que la référence particulière aux exigences spécifiées lors de la planification de la validation;
- les normes de référence;
- les outils et matériels utilisés, ainsi que les données d'étalonnage;
- les résultats de chaque essai;
- les écarts entre les résultats attendus et les résultats réels.

9.2 Analyse dans le cadre de la validation

9.2.1 Généralités

Le SCS doit être validé par une analyse. Les entrées de l'analyse incluent:

- la ou les fonctions de sécurité, ses/leurs caractéristiques et l'intégrité de sécurité spécifiée conformément à l'Article 5;
- la structure du système (les architectures de sous-système simple, par exemple) selon 7.5.2;
- les aspects quantifiables ($MTTF_D$ ou λ_D , DC et CCF , par exemple) selon 6.4.2;
- les aspects non quantifiables et qualitatifs qui ont un impact sur le comportement du système (le cas échéant, les aspects logiciels);
- les arguments déterministes.

NOTE 1 Un argument déterministe est un argument qui repose sur des aspects qualitatifs (qualité de fabrication, expérience d'utilisation, par exemple). La prise en considération dépend de l'application, qui, avec d'autres facteurs, peut avoir un impact sur les arguments déterministes.

NOTE 2 Les arguments déterministes diffèrent des autres preuves en ce sens qu'ils indiquent que les propriétés exigées du système découlent logiquement d'un modèle du système. Ces arguments peuvent être construits sur la base de concepts simples et bien compris.

9.2.2 Techniques d'analyse

Le choix d'une technique d'analyse dépend de l'objet particulier. Il existe deux techniques de base, qui sont les suivantes.

- a) Les techniques descendantes (déductives) permettent de déterminer les événements déclencheurs qui peuvent donner lieu à des événements majeurs identifiés, et de calculer la probabilité d'événements majeurs à partir de la probabilité d'événements déclencheurs. Elles peuvent également être utilisées pour déterminer les conséquences de plusieurs anomalies identifiées.

EXEMPLE Analyse par arbre de pannes (AAP, voir l'IEC 61025), analyse par arbre d'événements (AAE, voir l'IEC 62502).

- b) Les techniques ascendantes (inductives) permettent de déterminer les conséquences des premiers défauts identifiés.

EXEMPLE Analyse des modes de défaillance et de leurs effets (AMDE, voir l'IEC 60812) et analyse des modes de défaillance, de leurs effets et de la criticité (AMDEC).

9.2.3 Vérification de la spécification des exigences de sécurité (SRS)

La spécification des exigences pour la fonction de sécurité doit être vérifiée pour assurer la cohérence, l'exhaustivité et l'exactitude de son utilisation prévue.

La vérification peut être réalisée par des revues et des examens des exigences de sécurité et de la ou des spécifications de conception du SCS, en particulier pour démontrer que tous les aspects relatifs

- aux exigences d'application prévue et besoins de sécurité, et
- aux conditions opérationnelles et environnementales et possibles erreurs humaines (mauvaise utilisation, par exemple) ont été pris en considération.

9.3 Essais dans le cadre de la validation

9.3.1 Généralités

Des essais doivent être réalisés pour terminer la validation. Des essais de validation doivent être prévus et mis en œuvre de manière logique. En particulier:

- a) un plan d'essai doit être généré avant de commencer les essais. Il doit inclure
 - 1) les spécifications d'essai;
 - 2) le résultat exigé des essais pour la conformité, et
 - 3) la chronologie des essais;
- b) les enregistrements d'essai doivent être générés et doivent inclure
 - 4) le nom de la personne qui a procédé à l'essai;
 - 5) les conditions environnementales;
 - 6) les procédures d'essai et le matériel d'essai utilisé;
 - 7) la date de l'essai, et
 - 8) les résultats de l'essai;
- c) les enregistrements d'essai doivent être comparés au plan d'essai pour vérifier que les objectifs fonctionnels et de rendement ont été atteints.

L'échantillon d'essai doit être utilisé aussi proche que possible de sa configuration de fonctionnement finale.

Il peut s'agir d'un essai manuel ou automatique (par ordinateur, par exemple).

Si elle est appliquée, la validation des fonctions de sécurité par essai doit être réalisée en appliquant différentes combinaisons de signaux d'entrée au SCS. La réponse obtenue en sortie doit être comparée aux sorties appropriées spécifiées.

Il est recommandé d'appliquer systématiquement la combinaison de ces signaux d'entrée au système de commande et à la machine. La mise sous tension, le fonctionnement, les changements de direction et le redémarrage sont des exemples de cette logique. Le cas échéant, une plage étendue de données d'entrée doit être appliquée pour tenir compte des situations anormales ou inhabituelles, afin de voir comment le SCS répond. Ces combinaisons de données d'entrée doivent tenir compte des opérations incorrectes prévisibles.

Les objectifs de l'essai déterminent la condition environnementale de cet essai, qui peut être l'une l'autre des suivantes:

- les conditions environnementales de l'utilisateur prévu;
- les conditions à une caractéristique assignée particulière;
- une plage donnée de conditions, si une dérive est prévue.

9.3.2 Exactitude de mesure

L'exactitude des mesurages lors de la validation par essai doit être appropriée pour l'essai réalisé. En général, ces exactitudes de mesure doivent être dans les limites de 5 K pour les mesurages de température et de 5 % pour:

- a) les mesurages de temps;
- b) les mesurages de pression;

- c) les mesurages de force;
- d) les mesurages électriques;
- e) les mesurages d'humidité relative;
- f) les mesurages linéaires.

Les écarts par rapport à ces exactitudes de mesure doivent être justifiés.

9.3.3 Exigences plus strictes

Si, conformément à sa documentation d'accompagnement, les exigences du SCS dépassent celles du présent document, les exigences les plus strictes doivent s'appliquer.

NOTE Des exigences plus strictes peuvent s'appliquer si le système de commande doit résister à des conditions de service particulièrement défavorables (une manipulation brutale, les effets de l'humidité, l'hydrolyse, des variations de température ambiante, les effets de produits chimiques, la corrosion, des champs électromagnétiques de forte intensité, par exemple) dues, par exemple, à la proximité d'émetteurs.

9.3.4 Échantillons d'essai

Les échantillons d'essai ne doivent pas être modifiés au cours des essais.

Certains essais peuvent modifier définitivement les performances de certains composants. Si une modification définitive d'un composant empêche la partie relative à la sécurité de satisfaire aux exigences des essais supplémentaires, un nouvel échantillon (ou plusieurs) doit être utilisé pour les essais ultérieurs.

Si un essai particulier est destructif et que des résultats équivalents peuvent être obtenus en soumettant à l'essai la partie du SCS dans l'isolation, un échantillon de ce SCS peut être utilisé à la place du SCS dans son ensemble afin d'obtenir les résultats de l'essai. Cette approche doit uniquement être appliquée si une analyse a prouvé que l'essai d'une partie du SCS est suffisant pour démontrer l'intégrité de sécurité de l'ensemble du SCS qui réalise la fonction de sécurité.

9.4 Validation de la fonction de sécurité

9.4.1 Généralités

La validation des fonctions de sécurité doit démontrer que le SCS assure la ou les fonctions de sécurité selon leurs caractéristiques spécifiées.

NOTE 1 Une perte de la fonction de sécurité en l'absence d'anomalie matérielle est due à une anomalie systématique, qui peut être provoquée par des erreurs aux stades de la conception et de l'intégration (une mauvaise interprétation des caractéristiques de la fonction de sécurité, une erreur de conception logique, une erreur d'assemblage du matériel, une erreur de saisie du code du logiciel, etc.). Certaines de ces anomalies systématiques sont révélées lors du processus de conception, alors que d'autres le sont lors du processus de validation ou passent inaperçues. De plus, il est également possible de faire une erreur (défaillance de vérification d'une caractéristique, par exemple) lors du processus de validation.

Les caractéristiques spécifiées des fonctions de sécurité doivent être validées par l'application de mesures appropriées indiquées ci-après:

- analyse fonctionnelle des schémas et revues du logiciel (voir 9.5.3);

NOTE 2 Si les fonctions de sécurité d'une machine sont complexes ou nombreuses, une analyse peut réduire le nombre d'essais fonctionnels exigés.

- simulation;
- vérification des composants matériels installés dans la machine et détails des logiciels associés pour confirmer leur correspondance avec la documentation (fabrication, type, version, par exemple);

- essais fonctionnels des fonctions de sécurité dans tous les modes de fonctionnement exigés, tels que définis dans la SRS de la machine, afin d'établir si les caractéristiques spécifiées sont satisfaites (voir l'Article 5). Les essais fonctionnels doivent permettre d'assurer que toutes les sorties relatives à la sécurité sont obtenues sur leurs plages complètes et répondent aux signaux d'entrée relatifs à la sécurité conformément à la spécification. Les cas d'essai sont en principe déduits des spécifications, mais ils peuvent également inclure certains cas déduits de l'analyse des schémas ou des logiciels;
- essais fonctionnels étendus pour vérifier les signaux ou combinaisons de signaux anormaux/anormales prévisibles provenant d'une source d'entrée, y compris l'interruption et le rétablissement de l'alimentation, ainsi que les opérations incorrectes;
- vérification de l'aptitude à l'emploi de l'interface opérateur/SCS.

NOTE 3 Prendre en considération, par exemple, une IHM (interface homme-machine) pour la paramétrisation logicielle de la fonction de sécurité. En général, de plus amples informations sont disponibles dans l'IEC 60204-1 ou IEC 61310 (toutes les parties).

NOTE 4 D'autres mesures contre les défaillances systématiques mentionnées en 9.5.2 (diversité, détection des défaillances par des essais automatiques, par exemple) peuvent également contribuer à la détection des anomalies fonctionnelles.

9.4.2 Analyses et essais

Les analyses et essais exigent une analyse des défaillances à l'aide de schémas des circuits et, si l'analyse des défaillances ne donne pas de résultats clairs:

- des essais d'injection d'anomalie dans le circuit réel et déclenchement d'anomalie sur les composants réels, en particulier dans les parties du système dans lesquelles subsiste un doute quant aux résultats obtenus dans le cadre d'une analyse des défaillances (voir 9.2); ou
- une simulation du comportement du système de commande en cas d'anomalie, par exemple au moyen de modèles matériels et/ou logiciels.

Des essais d'injection d'anomalie ou de simulation d'anomalie peuvent être réalisés à différents niveaux (au niveau de l'élément de sous-système ou du sous-système, par exemple, en prenant en considération l'application et le montage d'essai spécifiques).

Lors de la validation par des essais, les essais doivent inclure, selon le cas,

- des essais d'injection d'anomalie dans un échantillon de production,
- des essais d'injection d'anomalie dans un modèle de matériel,
- une simulation logicielle des anomalies, et
- une défaillance du sous-système (alimentations électriques, par exemple).

L'instant précis auquel une anomalie est injectée dans un système peut être critique. L'effet le plus défavorable d'une injection d'anomalie doit être déterminé par une analyse et en injectant l'anomalie au moment critique approprié.

9.5 Validation de l'intégrité de sécurité du SCS

9.5.1 Généralités

Les étapes suivantes doivent être suivies:

- vérification de l'évaluation correcte du niveau SIL du SCS en fonction de l'architecture des sous-systèmes et des paramètres de fiabilité (DC et $MTTF_D$ ou λ_D , par exemple);
- vérification que le niveau SIL atteint par le SCS satisfait au niveau SIL exigé dans la spécification des exigences de sécurité pour les machines: $SIL \geq SIL \text{ exigé}$.

9.5.2 Validation du ou des sous-systèmes

L'intégrité de sécurité de chaque sous-système du SCS se caractérise par son niveau SIL et doit être validée en procédant à la confirmation (vérification) de ce qui suit:

- l'architecture utilisée (voir 7.5.2), et
- la *PFH* (voir 7.6), et
- l'intégrité systématique (voir 7.3.2, Logiciels, CCF).

Dans ce contexte, la $MTTF_D$ ou λ_D , la *DC* et la CCF sont validées par analyse et examen visuel. La vraisemblance des valeurs $MTTF_D$ ou λ_D des composants (y compris B_{10} ou B_{10D} , T_{10D} et les valeurs de cycle de fonctionnement) doit être vérifiée. Par exemple, la valeur donnée dans la fiche technique du fabricant doit être comparée à celle indiquée dans l'Annexe A.

NOTE 1 Une exclusion d'anomalie implique une valeur $MTTF_D$ infinie. Par conséquent, le composant ne contribue pas au calcul de la $MTTF_D$ du canal.

La vraisemblance des valeurs de couverture du diagnostic (*DC*) des composants (éléments de sous-système) et/ou des blocs logiques doit être vérifiée (par rapport aux mesures de l'Annex D, par exemple). La mise en œuvre correcte (matérielle et logicielle) des vérifications et des diagnostics, y compris la réaction appropriée à l'anomalie, doit être validée par des essais dans les conditions d'environnement classiques en utilisation.

La mise en œuvre correcte des mesures suffisantes contre les défaillances de cause commune doit être validée (par rapport à l'Annex E, par exemple). Les mesures de validation classiques sont l'analyse statique du matériel et les essais fonctionnels dans les conditions d'environnement.

NOTE 2 En règle générale, pour la spécification des valeurs de $MTTF_D$ ou λ_D des composants électroniques, une température ambiante de +40 °C sert de base. Lors de la validation, il est important de vérifier que, pour les valeurs de $MTTF_D$ ou λ_D , les conditions environnementales et fonctionnelles (en particulier la température) servant de base sont respectées. Si un dispositif ou un composant fonctionne bien au-dessus (plus de 15 °C) de la température spécifiée de +40 °C, il est nécessaire d'utiliser les valeurs de $MTTF_D$ ou λ_D pour la température ambiante augmentée.

9.5.3 Validation des mesures contre les défaillances systématiques

En règle générale, les mesures contre les défaillances systématiques peuvent être validées par:

- a) des examens des documents de conception qui confirment l'application
 - des principes de sécurité de base et des principes de sécurité éprouvés (voir l'ISO 13849-2:2012, Annexe A à Annexe D),
 - de mesures supplémentaires permettant d'éviter les défaillances systématiques, et
 - de mesures supplémentaires de maîtrise des défaillances systématiques (diversité du matériel, protection contre les modifications ou programmation d'assertion de défaillances, par exemple);
- b) analyse des défaillances (AMDE, par exemple);
- c) essais d'injection d'anomalie/déclenchement d'anomalie;
- d) examen et essai de la communication de données (paramétrisation, installation, par exemple);
- e) vérification qu'un système de gestion de la qualité permet d'éviter les causes de défaillances systématiques dans le processus de fabrication.

9.5.4 Validation du logiciel relatif à la sécurité

La validation du logiciel doit inclure:

- le comportement fonctionnel spécifié et les critères de performances (les performances de synchronisation, par exemple) du logiciel lorsqu'il est exécuté sur le matériel cible,
- la vérification que les mesures du logiciel sont suffisantes pour le niveau SIL exigé spécifié de la fonction de sécurité, et
- les mesures et activités réalisées lors du développement du logiciel pour éviter les anomalies logicielles systématiques.

Une première étape consiste à vérifier qu'il existe une documentation pour la spécification et la conception du logiciel relatif à la sécurité. Cette documentation doit être consultée pour confirmer son exhaustivité et l'absence d'interprétations erronées, d'omissions ou d'incohérences.

NOTE Dans le cas de petits programmes, il peut être suffisant d'analyser le programme par des examens ou dans le cadre d'une lecture croisée du flux de commande, des procédures, etc., à l'aide de la documentation du logiciel (organigramme de commande, code source des modules ou des blocs, listes d'allocations d'E/S et de variables, listes de références croisées).

En général, le logiciel peut être considéré comme une "boîte noire" ou une "boîte grise" (voir l'Article 8), puis être validé par l'essai boîte noire ou boîte grise, respectivement.

En fonction du niveau SIL exigé, il convient que les essais comprennent, selon le cas:

- des essais boîte noire du comportement fonctionnel et des performances (performances de synchronisation, par exemple),
- des cas d'essai étendus supplémentaires reposant sur des analyses de valeur limite, recommandés pour SIL 2 ou SIL 3,
- des essais d'E/S pour vérifier que les signaux d'entrée et de sortie relatifs à la sécurité sont correctement utilisés, et
- des cas d'essai qui simulent les anomalies préalablement déterminées de manière analytique, avec la réponse attendue, afin d'évaluer la pertinence des mesures logicielles pour la maîtrise des défaillances.

Chacune des fonctions logicielles qui a déjà été validée n'a pas besoin de l'être de nouveau. Toutefois, lorsqu'un certain nombre de blocs fonctionnels de sécurité sont combinés pour un projet particulier, l'ensemble de la fonction de sécurité obtenue doit être validée.

La documentation du logiciel doit être vérifiée pour confirmer que des mesures et activités suffisantes ont été mises en œuvre contre les anomalies logicielles systématiques conformément au modèle en V simplifié (voir la Figure 12).

Les mesures de gestion de la mise en œuvre, de la configuration et de la modification du logiciel selon l'Article 8, qui dépendent du niveau SIL à atteindre, doivent être examinées en ce qui concerne leur mise en œuvre correcte.

S'il convient d'apporter des modifications importantes au logiciel relatif à la sécurité, elles doivent être de nouveau validées sur une échelle appropriée.

9.5.5 Validation de la combinaison de sous-systèmes

Lorsque la fonction de sécurité est mise en œuvre par au moins deux sous-systèmes, la validation de la combinaison (par analyse et, le cas échéant, par des essais) doit être réalisée pour établir que la combinaison atteint l'intégrité de sécurité spécifiée dans la conception. Les résultats de validation enregistrés des sous-systèmes peuvent être pris en compte. Les étapes de validation suivantes doivent être suivies:

- examen des documents de conception décrivant la ou les fonctions de sécurité globales;
- une vérification de l'évaluation correcte du niveau SIL global de la combinaison de sous-systèmes, en fonction du niveau SIL de chaque sous-système individuel (voir 6.4.2);
- prise en considération des caractéristiques des interfaces (tension, courant, pression, format de données des informations, niveau du signal, par exemple);

- analyse des défaillances liées à la combinaison/l'intégration (par une AMDE, par exemple);
- pour les systèmes redondants, essais d'injection d'anomalie en relation avec la combinaison/l'intégration.

10 Documentation

10.1 Généralités

Le fabricant d'un SCS et celui des sous-systèmes doivent élaborer une documentation technique pertinente (voir 10.2) et des informations d'utilisation (voir 10.3).

La documentation doit présenter la procédure qui a été suivie et les résultats qui ont été obtenus.

La documentation doit faire l'objet d'un contrôle de version.

10.2 Documentation technique

La documentation doit contenir des informations pertinentes pour la partie relative à la sécurité:

- la ou les fonctions de sécurité fournies par le SCS selon l'Article 5 ou la sous-fonction de sécurité fournie par le sous-système SCS;

NOTE 1 Seules les fonctions de sécurité exigées par l'application particulière nécessitent d'être prises en considération.

- Si la conception inclut la conception du sous-système (voir l'Article 7), la documentation technique doit:
 - couvrir l'essai ou l'analyse du comportement sous anomalie donnant lieu à une perte de la fonction de sécurité ou
 - donner un exemple qualifié (une note d'application, par exemple);
- les caractéristiques de chaque fonction de sécurité selon 5.2;
- les mesures d'essai périodique lorsque les essais périodiques sont définis pour le SCS;
- les conditions environnementales;
- les mesures contre les défaillances systématiques (dans le cadre de règles de conception génériques réalisées par des éléments dans les limites du document d'appréciation du risque, par exemple);
- la documentation du logiciel selon l'Article 8;

NOTE 2 En général, cette documentation est destinée à répondre aux besoins internes du fabricant, et n'est pas transmise à l'utilisateur de la machine.

Si des composants éprouvés sont utilisés, la documentation de ces composants doit inclure les aspects suivants:

- description de la version, du composant et de l'application,
- informations spécifiques à l'application
 - limites d'utilisation du composant à considérer comme étant éprouvé,
 - analyse de pertinence: par exemple, comportement fonctionnel, exactitude, comportement en cas d'anomalie, réponse temporelle, aptitude à l'emploi et maintenabilité,
 - essais exigés,
- lorsque, en fonction de l'utilisation passée pour démontrer l'équivalence entre le fonctionnement prévu et l'expérience de fonctionnement précédente, une analyse d'impact sur les différences entre un cas d'utilisation passé et une situation actuelle doit être effectuée.

Le Tableau 9 récapitule la documentation à fournir, si nécessaire.

Tableau 9 – Documentation d'un SCS

Informations exigées	Paragraphe
Plan de sécurité fonctionnelle	4.3
Spécification des exigences de sécurité (SRS)	5.2
Spécification des exigences fonctionnelles (pour le SCS)	5.2.3
Spécification des exigences d'intégrité de sécurité (pour le	5.2.5
Conception du SCS	6.3
Processus de conception structuré	4.2
Structure des sous-fonctions	6.3.2
Architecture du SCS	6.3.2
Sous-fonction et exigences de sécurité du sous-système	6.3.3
Réalisation du sous-système	7.1
Architecture du sous-système	7.2
Exclusions d'anomalies revendiquées lors de l'estimation	7.3.3.3 et 7.4.1
Ensemble sous-système	7.4 et 7.5
Exigences de sécurité du logiciel	8.3.2.2 ou 8.4.2.2
Paramétrisation liée au logiciel	6.7.5
Éléments de gestion de configuration du logiciel	8.3.8 ou 8.4.10
Adaptabilité des outils de développement du logiciel	8.3.1.3 ou 8.4.1.3
Documentation du programme d'application	8.3.7 ou 8.4.9
Résultats des essais du module logiciel d'application	8.3.5 ou 8.4.6
Résultats des essais d'intégration du logiciel d'application	8.4.7
Documentation sur l'intégration du SCS (essais)	9.5.4
Documentation des composants éprouvés	10.2
Documentation pour l'installation, l'utilisation et l'entretien	10.3
Documentation sur les essais de validation du SCS	9.4 et 9.5
Documentation pour la gestion de configuration du SCS	4.4

Voir l'Annex I, qui donne des exemples d'activités, de documents et de rôles.

10.3 Informations pour l'utilisation du SCS

10.3.1 Généralités

10.3.1.1 Vue d'ensemble

Les informations relatives à l'utilisation du SCS doivent porter sur l'installation, l'utilisation et la maintenance. Elles doivent inclure une description détaillée du matériel, de son installation et de son montage, comme suit.

10.3.1.2 Spécification de l'intégrité de sécurité

Des informations particulières doivent être données sur l'intégrité de sécurité du SCS, comme suit:

- SIL 1, 2 ou 3,
- le cas échéant, les contraintes architecturales du ou des sous-systèmes.

10.3.1.3 SCS et sous-systèmes

Les SCS sont en général conçus et mis en œuvre comme un système relatif à la sécurité par le fabricant d'une machine à l'aide de sous-systèmes séparés disponibles.

Les sous-systèmes sont en général fabriqués et commercialisés comme un dispositif complet prêt à l'emploi.

Par conséquent, des exigences différentes en matière d'informations relatives à l'utilisation s'appliquent au fabricant de la machine ou à celui des sous-systèmes. Le fabricant d'une machine peut également jouer le rôle d'un fabricant de sous-systèmes SCS.

10.3.2 Informations relatives à l'utilisation données par le fabricant de sous-systèmes

Les principes de l'ISO 12100:2010, 6.4, et les sections applicables d'autres documents pertinents (l'IEC 60204-1:2016, Article 17, par exemple) doivent s'appliquer.

En particulier, le fabricant d'un sous-système doit indiquer, dans ses instructions, les informations importantes pour assurer une installation, une utilisation et une maintenance en toute sécurité du sous-système. Cela doit inclure, entre autres, ce qui suit:

- a) une description du sous-système, y compris:
 - une description générale du sous-système et de sa fonction;
 - des instructions d'installation;
 - les exigences d'interface;
 - des informations relatives à la configuration, aux réglages ou à la programmation et, le cas échéant, une mention de l'usage prévu du sous-système, et toutes les mesures qui peuvent se révéler nécessaires pour empêcher un mauvais usage raisonnablement prévisible;
- b) des informations relatives aux limites d'exploitation du sous-système, y compris:
 - la spécification des limites environnementales (température, luminosité, vibrations, bruit, polluants atmosphériques, par exemple);
 - la spécification des limites d'interface (caractéristiques électriques, hydrauliques, pneumatiques ou mécaniques, par exemple);
 - la spécification d'autres limites pertinentes pour la fonctionnalité de sécurité prévue (fréquence de fonctionnement, intensité, plage, par exemple);
- c) une description des exclusions d'anomalie essentielles pour le maintien de l'intégrité de sécurité prévue. Des informations appropriées (pour la modification, la maintenance et la réparation, par exemple) doivent être données pour assurer la justification continue de l'exclusion ou des exclusions d'anomalie;
- d) une description de toutes les mesures nécessaires au niveau du sous-système pour garantir l'absence de dégradation de la fonction SCS prévue provoquée par le système de commande d'une machine;
- e) le temps de réponse du sous-système;
- f) la durée de fonctionnement utile du sous-système;
- g) des informations relatives aux fonctions de diagnostic exigées pour l'interface correcte et la sécurité d'utilisation;
- h) des informations relatives aux indications et alarmes;
- i) la nature et la fréquence des procédures d'examen exigées;
- j) la nature et la fréquence des procédures d'essai exigées (essais, par exemple) si le diagnostic est toujours en cours;
- k) les dispositions en matière de maintenabilité du sous-système, le cas échéant. Toutes les informations relatives à la maintenance doivent satisfaire à l'ISO 12100:2010, 6.4.5.1 e). Les informations doivent inclure:
 - les procédures de diagnostic et de réparation des anomalies;
 - les procédures pour confirmer le fonctionnement correct après réparation;
- l) les paramètres relatifs à la sécurité (*PFH*, *PF*, *SIL*, par exemple).

10.3.3 Informations relatives à l'utilisation données par l'intégrateur du SCS

L'intégrateur SCS (en général le fabricant de la machine) doit inclure les informations pertinentes dans les instructions d'utilisation pour permettre à l'utilisateur de la machine de développer des procédures visant à maintenir la sécurité fonctionnelle exigée du SCS pendant l'utilisation et la maintenance de la machine.

En particulier, l'intégrateur SCS doit indiquer, dans les instructions, les informations importantes pour la sécurité d'utilisation du SCS, y compris celles relatives aux mesures qui peuvent s'avérer nécessaires pour éviter toute mauvaise utilisation raisonnablement prévisible.

Ces informations d'utilisation doivent inclure, entre autres, les suivantes:

- a) les limites de fonctionnement du SCS (y compris les conditions environnementales);
- b) des descriptions claires et les instructions connexes pour les interfaces utilisateur avec le SCS (le panneau de commande de l'opérateur, les indications et alarmes, par exemple);
- c) une description des fonctions de sécurité mises en œuvre dans le SCS, y compris une description des phénomènes dangereux et des situations dangereuses, le mode de fonctionnement à sollicitation, l'état de sécurité, le temps de sécurité du processus, un aperçu du ou des schémas (fonctionnels) et du ou des schémas des circuits, selon le cas;
- d) une description (y compris les schémas d'interconnexion) de l'interaction (s'il y a lieu) entre la ou les fonctions du SCS et la ou les fonctions du système de commande de la machine;
- e) le marquage, si cela est exigé, selon l'ISO 12100:2010, 6.4.4;
- f) la durée de fonctionnement utile et les exigences relatives aux composants SCS;
- g) les informations relatives à une inhibition et/ou suspension des fonctions de sécurité;
- h) un mode de fonctionnement correspondant à la ou aux fonctions de sécurité;
- i) un examen et des essais périodiques, le cas échéant (une certaine distance de sécurité doit être soumise à l'essai régulièrement, par exemple), y compris la nature de l'une des procédures d'essai exigées (voir également 6.9 pour des détails);
- j) les outils nécessaires à la maintenance et à la remise en service, ainsi que les procédures de maintenance des outils et des équipements;
- k) les dispositions en matière de maintenance du SCS, le cas échéant, y compris les implications pour l'exclusion d'anomalie. Toutes les informations relatives à la maintenance doivent satisfaire à l'ISO 12100:2010, 6.4.5.1 e). Les informations doivent inclure:
 - les procédures de diagnostic et de réparation des anomalies (instructions pour le rétablissement fonctionnel du SCS en cas de défaillance, par exemple);
 - les procédures pour confirmer le fonctionnement correct après réparation;
 - l'entretien préventif et l'entretien correctif.

NOTE 1 Les essais périodiques sont des essais fonctionnels nécessaires pour confirmer le bon fonctionnement et détecter les anomalies.

NOTE 2 L'entretien préventif désigne les mesures prises pour maintenir le fonctionnement exigé du SCS.

NOTE 3 L'entretien correctif désigne les mesures prises après l'apparition d'une ou de plusieurs anomalies spécifiques qui permettent au SCS de retourner dans l'état "tel qu'il a été conçu".