IEC 61508-3

Edition 2.0 2010-04

INTERNATIONAL STANDARD

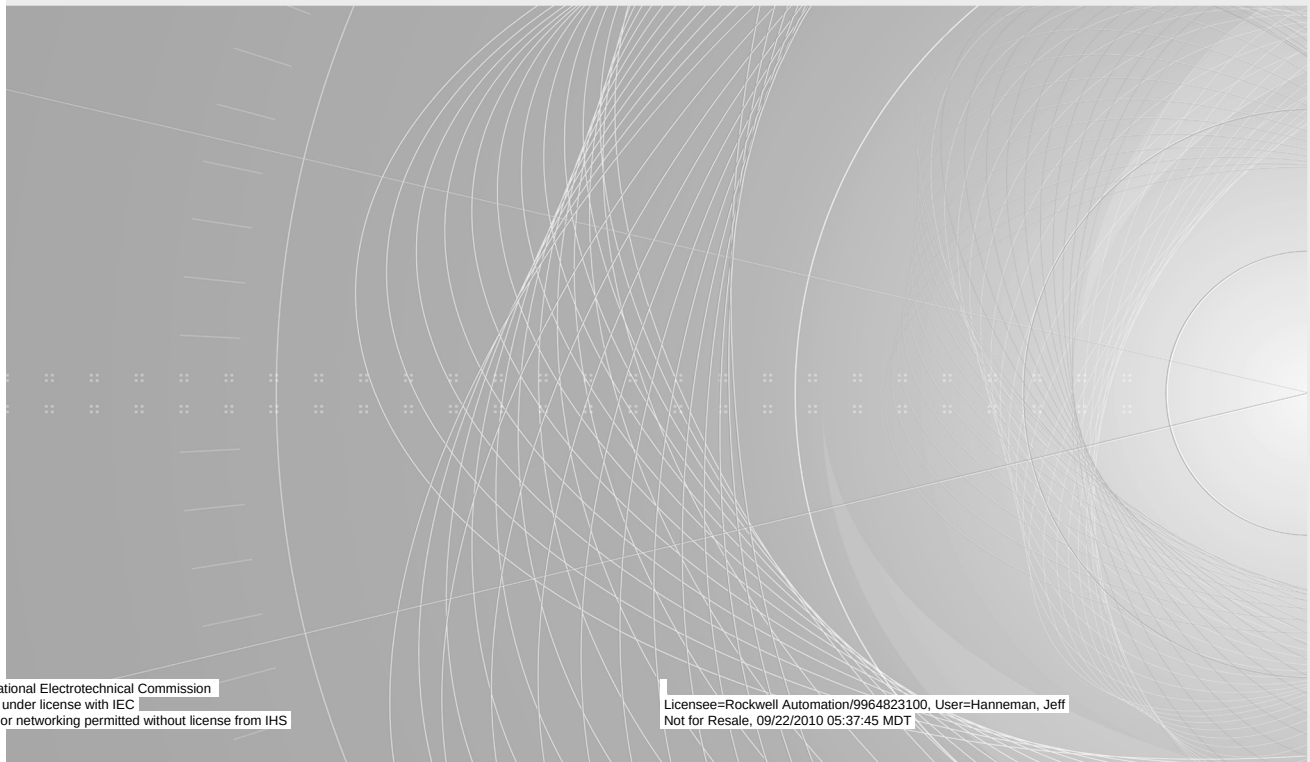
NORME INTERNATIONALE

BASIC SAFETY PUBLICATION

PUBLICATION FONDAMENTALE DE SÉCURITÉ

Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements

Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité – Partie 3: Exigences concernant les logiciels



IEC 61508-3:2010

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2010 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 la CEI ou du Comité national de la CEI du pays du demandeur.
Si vous avez des questions sur le copyright de la CEI ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de la CEI de votre pays de résidence.

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: 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 corrigenda or an amendment might have been published.

- Catalogue of IEC publications: www.iec.ch/searchpub

The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub

Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv

If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:
Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

A propos de la CEI

La Commission Electrotechnique Internationale (CEI) 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 CEI

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

- Catalogue des publications de la CEI: www.iec.ch/searchpub/cur_fut-f.htm

Le Catalogue en-ligne de la CEI vous permet d'effectuer des recherches en utilisant différents critères (numéro de référence, texte, comité d'études,…). Il donne aussi des informations sur les projets et les publications retirées ou remplacées.

- Just Published CEI: www.iec.ch/online_news/justpub

Restez informé sur les nouvelles publications de la CEI. Just Published détaille deux fois par mois les nouvelles publications parues. Disponible en-ligne et aussi par email.

- Electropedia: www.electropedia.org

Le premier dictionnaire en ligne au monde de termes électroniques et électriques. Il contient plus de 20 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans les langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International en ligne.

- Service Clients: www.iec.ch/webstore/custserv/custserv_entry-f.htm

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions, visitez le FAQ du Service clients ou contactez-nous:
Email: csc@iec.ch
Tél.: +41 22 919 02 11
Fax: +41 22 919 03 00

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

IEC 61508-3

Edition 2.0 2010-04

INTERNATIONAL STANDARD

NORME INTERNATIONALE

BASIC SAFETY PUBLICATION

PUBLICATION FONDAMENTALE DE SÉCURITÉ

Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements

Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité – Partie 3: Exigences concernant les logiciels

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMISSION ELECTROTECHNIQUE INTERNATIONALE

PRICE CODE XE
CODE PRIX

ICS 25.040.40

ISBN 978-2-88910-526-7

® Registered trademark of the International Electrotechnical Commission
® de la Commission Electrotechnique Internationale

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

CONTENTS

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

--`,,``,,`````````,,``,`,,``,`,`-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FUNCTIONAL SAFETY OF ELECTRICAL/ELECTRONIC/ PROGRAMMABLE ELECTRONIC SAFETY-RELATED SYSTEMS –

Part 3: Software requirements

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.

International Standard IEC 61508-3 has been prepared by subcommittee 65A: System aspects, of IEC technical committee 65: Industrial-process measurement, control and automation.

This second edition cancels and replaces the first edition published in 1998. This edition constitutes a technical revision.

This edition has been subject to a thorough review and incorporates many comments received at the various revision stages.

It has the status of a basic safety publication according to IEC Guide 104.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

The text of this standard is based on the following documents:

FDIS	Report on voting
65A/550/FDIS	65A/574/RVD

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

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts of the IEC 61508 series, published under the general title *Functional safety of electrical / electronic / programmable electronic safety-related systems*, can be found on the IEC website.

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.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

INTRODUCTION

Systems comprised of electrical and/or electronic elements have been used for many years to perform safety functions in most application sectors. Computer-based systems (generically referred to as programmable electronic systems) are being used in all application sectors to perform non-safety functions and, increasingly, to perform safety functions. If computer system technology is to be effectively and safely exploited, it is essential that those responsible for making decisions have sufficient guidance on the safety aspects on which to make these decisions.

This International Standard sets out a generic approach for all safety lifecycle activities for systems comprised of electrical and/or electronic and/or programmable electronic (E/E/PE) elements that are used to perform safety functions. This unified approach has been adopted in order that a rational and consistent technical policy be developed for all electrically-based safety-related systems. A major objective is to facilitate the development of product and application sector international standards based on the IEC 61508 series.

NOTE 1 Examples of product and application sector international standards based on the IEC 61508 series are given in the bibliography (see references [1], [2] and [3]).

In most situations, safety is achieved by a number of systems which rely on many technologies (for example mechanical, hydraulic, pneumatic, electrical, electronic, programmable electronic). Any safety strategy must therefore consider not only all the elements within an individual system (for example sensors, controlling devices and actuators) but also all the safety-related systems making up the total combination of safety-related systems. Therefore, while this International Standard is concerned with E/E/PE safety-related systems, it may also provide a framework within which safety-related systems based on other technologies may be considered.

It is recognized that there is a great variety of applications using E/E/PE safety-related systems in a variety of application sectors and covering a wide range of complexity, hazard and risk potentials. In any particular application, the required safety measures will be dependent on many factors specific to the application. This International Standard, by being generic, will enable such measures to be formulated in future product and application sector international standards and in revisions of those that already exist.

This International Standard

– considers all relevant overall, E/E/PE system and software safety lifecycle phases (for example, from initial concept, through design, implementation, operation and maintenance to decommissioning) when E/E/PE systems are used to perform safety functions;
– has been conceived with a rapidly developing technology in mind; the framework is sufficiently robust and comprehensive to cater for future developments;
– enables product and application sector international standards, dealing with E/E/PE safety-related systems, to be developed; the development of product and application sector international standards, within the framework of this standard, should lead to a high level of consistency (for example, of underlying principles, terminology etc.) both within application sectors and across application sectors; this will have both safety and economic benefits;
– provides a method for the development of the safety requirements specification necessary to achieve the required functional safety for E/E/PE safety-related systems;
– adopts a risk-based approach by which the safety integrity requirements can be determined;
– introduces safety integrity levels for specifying the target level of safety integrity for the safety functions to be implemented by the E/E/PE safety-related systems;

NOTE 2 The standard does not specify the safety integrity level requirements for any safety function, nor does it mandate how the safety integrity level is determined. Instead it provides a risk-based conceptual framework and example techniques.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

– sets target failure measures for safety functions carried out by E/E/PE safety-related systems, which are linked to the safety integrity levels;

– sets a lower limit on the target failure measures for a safety function carried out by a single E/E/PE safety-related system. For E/E/PE safety-related systems operating in

  – a low demand mode of operation, the lower limit is set at an average probability of a dangerous failure on demand of 10^{-5};

  – a high demand or a continuous mode of operation, the lower limit is set at an average frequency of a dangerous failure of 10^{-9} [h^{-1}];

NOTE 3 A single E/E/PE safety-related system does not necessarily mean a single-channel architecture.

NOTE 4 It may be possible to achieve designs of safety-related systems with lower values for the target safety integrity for non-complex systems, but these limits are considered to represent what can be achieved for relatively complex systems (for example programmable electronic safety-related systems) at the present time.

– sets requirements for the avoidance and control of systematic faults, which are based on experience and judgement from practical experience gained in industry. Even though the probability of occurrence of systematic failures cannot in general be quantified the standard does, however, allow a claim to be made, for a specified safety function, that the target failure measure associated with the safety function can be considered to be achieved if all the requirements in the standard have been met;

– introduces systematic capability which applies to an element with respect to its confidence that the systematic safety integrity meets the requirements of the specified safety integrity level;

– adopts a broad range of principles, techniques and measures to achieve functional safety for E/E/PE safety-related systems, but does not explicitly use the concept of fail safe. However, the concepts of “fail safe” and “inherently safe” principles may be applicable and adoption of such concepts is acceptable providing the requirements of the relevant clauses in the standard are met.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

FUNCTIONAL SAFETY OF ELECTRICAL/ELECTRONIC/ PROGRAMMABLE ELECTRONIC SAFETY-RELATED SYSTEMS –

Part 3: Software requirements

1 Scope

1.1 This part of the IEC 61508 series

a) is intended to be utilized only after a thorough understanding of IEC 61508-1 and IEC 61508-2;

b) applies to any software forming part of a safety-related system or used to develop a safety-related system within the scope of IEC 61508-1 and IEC 61508-2. Such software is termed safety-related software (including operating systems, system software, software in communication networks, human-computer interface functions, and firmware as well as application software);

c) provides specific requirements applicable to support tools used to develop and configure a safety-related system within the scope of IEC 61508-1 and IEC 61508-2;

d) requires that the software safety functions and software systematic capability are specified;

NOTE 1 If this has already been done as part of the specification of the E/E/PE safety-related systems (see 7.2 of IEC 61508-2), then it does not have to be repeated in this part.

NOTE 2 Specifying the software safety functions and software systematic capability is an iterative procedure; see Figures 3 and 6.

NOTE 3 See Clause 5 and Annex A of IEC 61508-1 for documentation structure. The documentation structure may take account of company procedures, and of the working practices of specific application sectors.

NOTE 4 Note: See 3.5.9 of IEC 61508-4 for definition of the term "systematic capability".

e) establishes requirements for safety lifecycle phases and activities which shall be applied during the design and development of the safety-related software (the software safety lifecycle model). These requirements include the application of measures and techniques, which are graded against the required systematic capability, for the avoidance of and control of faults and failures in the software;

f) provides requirements for information relating to the software aspects of system safety validation to be passed to the organisation carrying out the E/E/PE system integration;

g) provides requirements for the preparation of information and procedures concerning software needed by the user for the operation and maintenance of the E/E/PE safety-related system;

h) provides requirements to be met by the organisation carrying out modifications to safety-related software;

i) provides, in conjunction with IEC 61508-1 and IEC 61508-2, requirements for support tools such as development and design tools, language translators, testing and debugging tools, configuration management tools;

NOTE 4 Figure 5 shows the relationship between IEC 61508-2 and IEC 61508-3.

j) Does not apply for medical equipment in compliance with the IEC 60601 series.

1.2 IEC 61508-1, IEC 61598-2, IEC 61508-3 and IEC 61508-4 are basic safety publications, although this status does not apply in the context of low complexity E/E/PE safety-related systems (see 3.4.3 of IEC 61508-4). As basic safety publications, they are intended for use by technical committees in the preparation of standards in accordance with the principles contained in IEC Guide 104 and ISO/IEC Guide 51. IEC 61508-1, IEC 61508-2, IEC 61508-3 and IEC 61508-4 are also intended for use as stand-alone publications. The horizontal safety

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

function of this international standard does not apply to medical equipment in compliance with the IEC 60601 series.

1.3 One of the responsibilities of a technical committee is, wherever applicable, to make use of basic safety publications in the preparation of its publications. In this context, the requirements, test methods or test conditions of this basic safety publication will not apply unless specifically referred to or included in the publications prepared by those technical committees.

1.4 Figure 1 shows the overall framework of the IEC 61508 series and indicates the role that IEC 61508-3 plays in the achievement of functional safety for E/E/PE safety-related systems.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

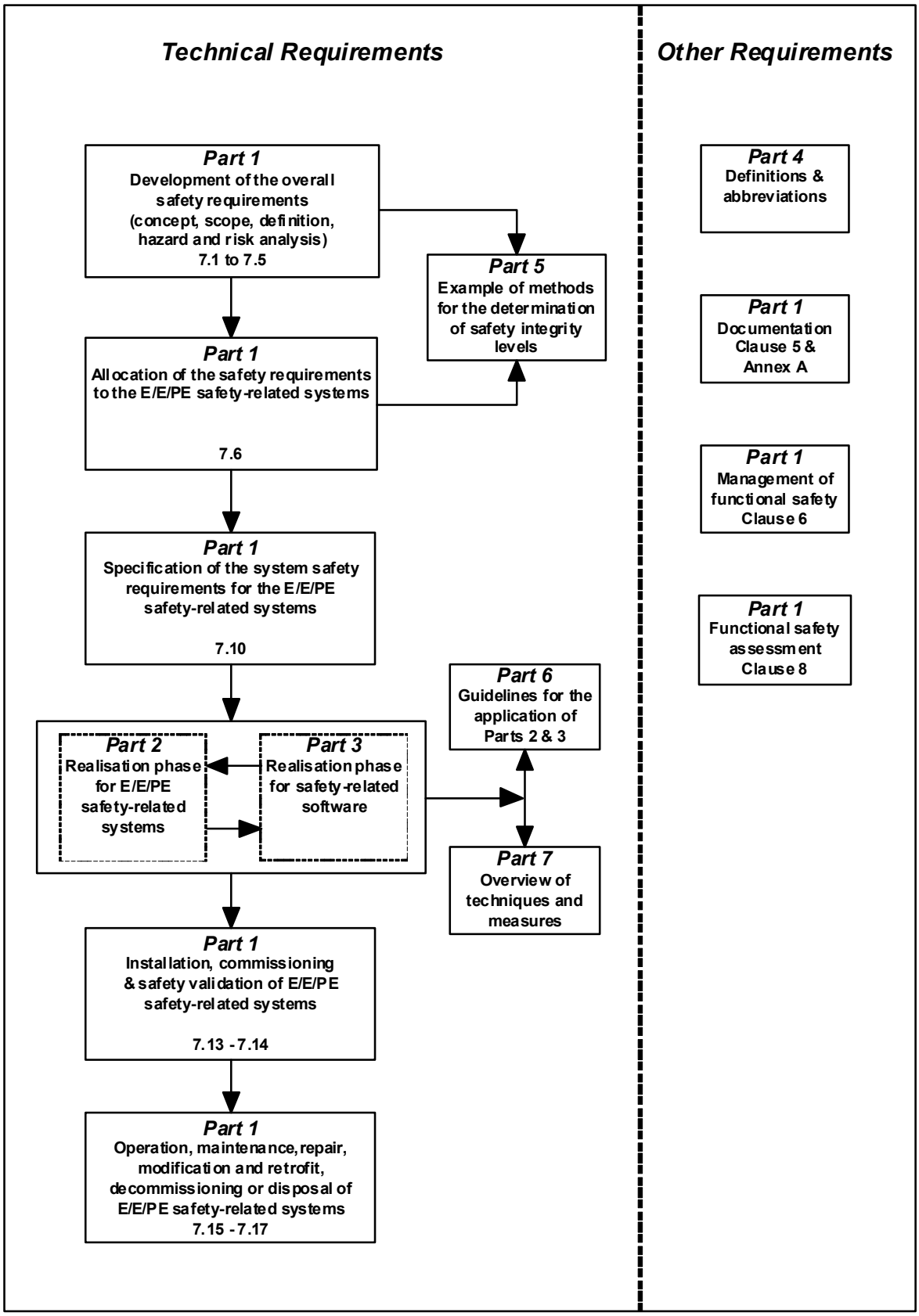


Figure 1 – Overall framework of the IEC 61508 series

--`,,```,,,,````-`-`,,`,,`,`,,`---
Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

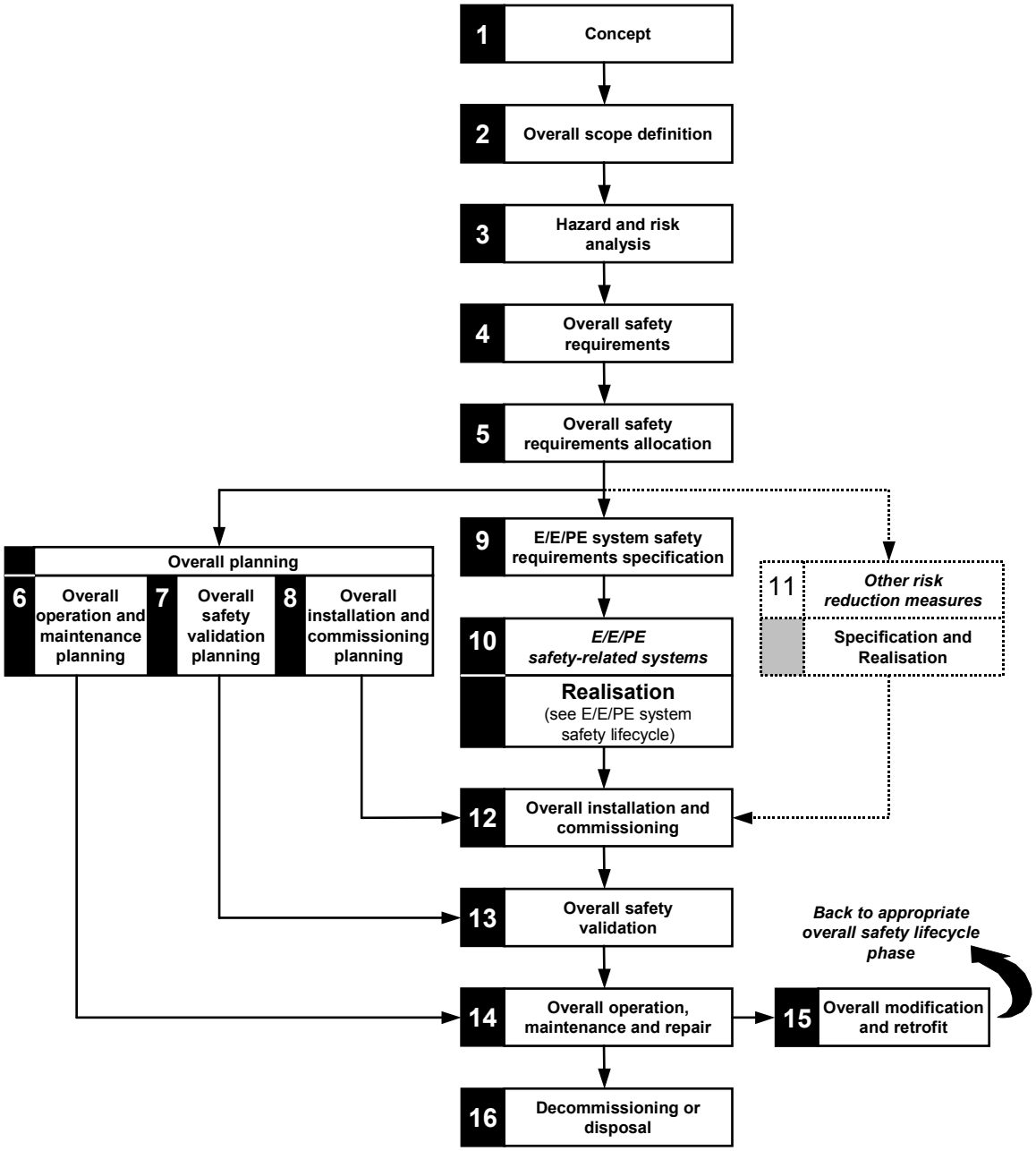


Figure 2 – Overall safety lifecycle

2 Normative references

The following referenced documents are indispensable for the application 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 61508-1: 2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements*

--`,``,,`````````,,``,`,,``,`,`-`-`,,`,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

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 61508-4: 2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 4: Definitions and abbreviations*

IEC Guide 104:1997, *The preparation of safety publications and the use of basic safety publications and group safety publications*

IEC/ISO Guide 51:1999, *Safety aspects – Guidelines for their inclusion in standards*

3 Definitions and abbreviations

For the purposes of this document, the definitions and abbreviations given in IEC 61508-4 apply.

4 Conformance to this standard

The requirements for conformance to this standard are given in Clause 4 of IEC 61508-1.

5 Documentation

The objectives and requirements for documentation are given in Clause 5 of IEC 61508-1.

6 Additional requirements for management of safety-related software

6.1 Objectives

The objectives are as detailed in 6.1 of IEC 61508-1.

6.2 Requirements

6.2.1 The requirements are as detailed in 6.2 of IEC 61508-1, with the following additional requirements.

6.2.2 The functional safety planning shall define the strategy for software procurement, development, integration, verification, validation and modification to the extent required by the safety integrity level of the safety functions implemented by the E/E/PE safety-related system.

NOTE The philosophy of this approach is to use the functional safety planning as an opportunity to customize this standard to take account of the required safety integrity for each safety function implemented by the E/E/PE safety-related system.

6.2.3 Software configuration management shall:

a) apply administrative and technical controls throughout the software safety lifecycle, in order to manage software changes and thus ensure that the specified requirements for safety-related software continue to be satisfied;

b) guarantee that all necessary operations have been carried out to demonstrate that the required software systematic capability has been achieved;

c) maintain accurately and with unique identification all configuration items which are necessary to meet the safety integrity requirements of the E/E/PE safety-related system. Configuration items include at least the following: safety analysis and requirements; software specification and design documents; software source code modules; test plans

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

and results; verification documents; pre-existing software elements and packages which are to be incorporated into the E/E/PE safety-related system; all tools and development environments which are used to create or test, or carry out any action on, the software of the E/E/PE safety-related system;

d) apply change-control procedures:

- to prevent unauthorized modifications; to document modification requests;
- to analyse the impact of a proposed modification, and to approve or reject the request;
- to document the details of, and the authorisation for, all approved modifications;
- to establish configuration baseline at appropriate points in the software development, and to document the (partial) integration testing of the baseline;
- to guarantee the composition of, and the building of, all software baselines (including the rebuilding of earlier baselines).

NOTE 1 Management decision and authority is needed to guide and enforce the use of administrative and technical controls.

NOTE 2 At one extreme, an impact analysis may include an informal assessment. At the other extreme, an impact analysis may include a rigorous formal analysis of the potential adverse impact of all proposed changes which may be inadequately understood or implemented. See IEC 61508-7 for guidance on impact analysis.

e) ensure that appropriate methods are implemented to load valid software elements and data correctly into the run-time system;

NOTE 3 This may include consideration of specific target location systems as well as general systems. Software other than application might need a safe loading method, e.g. firmware.

f) document the following information to permit a subsequent functional safety audit: configuration status, release status, the justification (taking account of the impact analysis) for and approval of all modifications, and the details of the modification;

g) formally document the release of safety-related software. Master copies of the software and all associated documentation and version of data in service shall be kept to permit maintenance and modification throughout the operational lifetime of the released software.

NOTE 4 For further information on configuration management, see IEC 61508-7

7 Software safety lifecycle requirements

7.1 General

7.1.1 Objective

The objective of the requirements of this subclause is to structure the development of the software into defined phases and activities (see Table 1 and Figures 3 to 6).

7.1.2 Requirements

7.1.2.1 A safety lifecycle for the development of software shall be selected and specified during safety planning in accordance with Clause 6 of IEC 61508-1.

7.1.2.2 Any software lifecycle model may be used provided all the objectives and requirements of this clause are met.

7.1.2.3 Each phase of the software safety lifecycle shall be divided into elementary activities with the scope, inputs and outputs specified for each phase.

NOTE See Figures 3, 4 and Table 1.

7.1.2.4 Provided that the software safety lifecycle satisfies the requirements of Table 1, it is acceptable to tailor the V-model (see Figure 6) to take account of the safety integrity and the complexity of the project.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

NOTE 1 A software safety lifecycle model which satisfies the requirements of this clause may be suitably customized for the particular needs of the project or organisation. The full list of lifecycle phases in Table 1 is suitable for large newly developed systems. In small systems, it might be appropriate, for example, to merge the phases of software system design and architectural design.

NOTE 2 See Annex G for the characteristics of data-driven systems (e.g. full variability / limited variability programming languages, extent of data configuration) that may be relevant when customising the software safety lifecycle.

7.1.2.5 Any customisation of the software safety lifecycle shall be justified on the basis of functional safety.

7.1.2.6 Quality and safety assurance procedures shall be integrated into safety lifecycle activities.

7.1.2.7 For each lifecycle phase, appropriate techniques and measures shall be used. Annexes A and B provide a guide to the selection of techniques and measures, and references to IEC 61508-6 and IEC 61508-7. IEC 61508-6 and IEC 61508-7 give recommendations on specific techniques to achieve the properties required for systematic safety integrity. Selecting techniques from these recommendations does not guarantee by itself that the required safety integrity will be achieved.

NOTE Success in achieving systematic safety integrity depends on selecting techniques with attention to the following factors:

– the consistency and the complementary nature of the chosen methods, languages and tools for the whole development cycle;

– whether the developers use methods, languages and tools they fully understand;

– whether the methods, languages and tools are well-adapted to the specific problems encountered during development.

7.1.2.8 The results of the activities in the software safety lifecycle shall be documented (see Clause 5).

NOTE Clause 5 of IEC 61508-1 considers the documented outputs from the safety lifecycle phases. In the development of some E/E/PE safety-related systems, the output from some safety lifecycle phases may be a distinct document, while the documented outputs from several phases may be merged. The essential requirement is that the output of the safety lifecycle phase be fit for its intended purpose.

7.1.2.9 If at any phase of the software safety lifecycle, a modification is required pertaining to an earlier lifecycle phase, then an impact analysis shall determine (1) which software modules are impacted, and (2) which earlier safety lifecycle activities shall be repeated.

NOTE At one extreme, an impact analysis may include an informal assessment. At the other extreme, an impact analysis may include a rigorous formal analysis of the potential adverse impact of all proposed changes which may be inadequately understood or implemented. See IEC 61508-7 for guidance on impact analysis.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

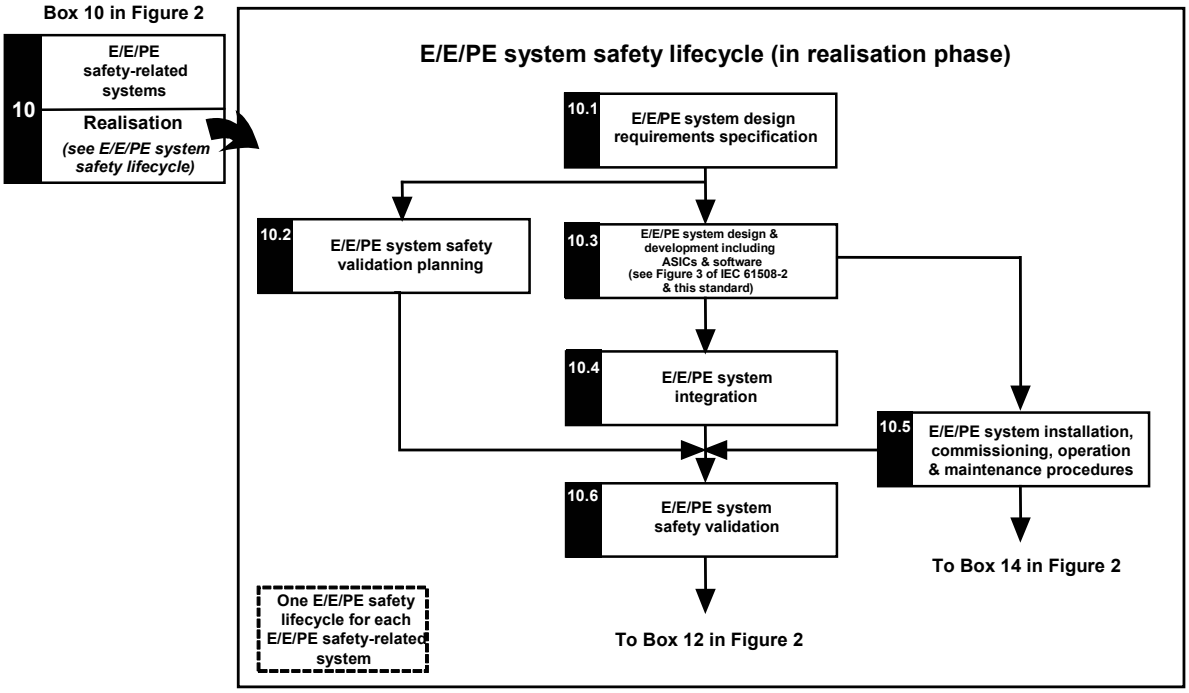


Figure 3 – E/E/PE system safety lifecycle (in realisation phase)

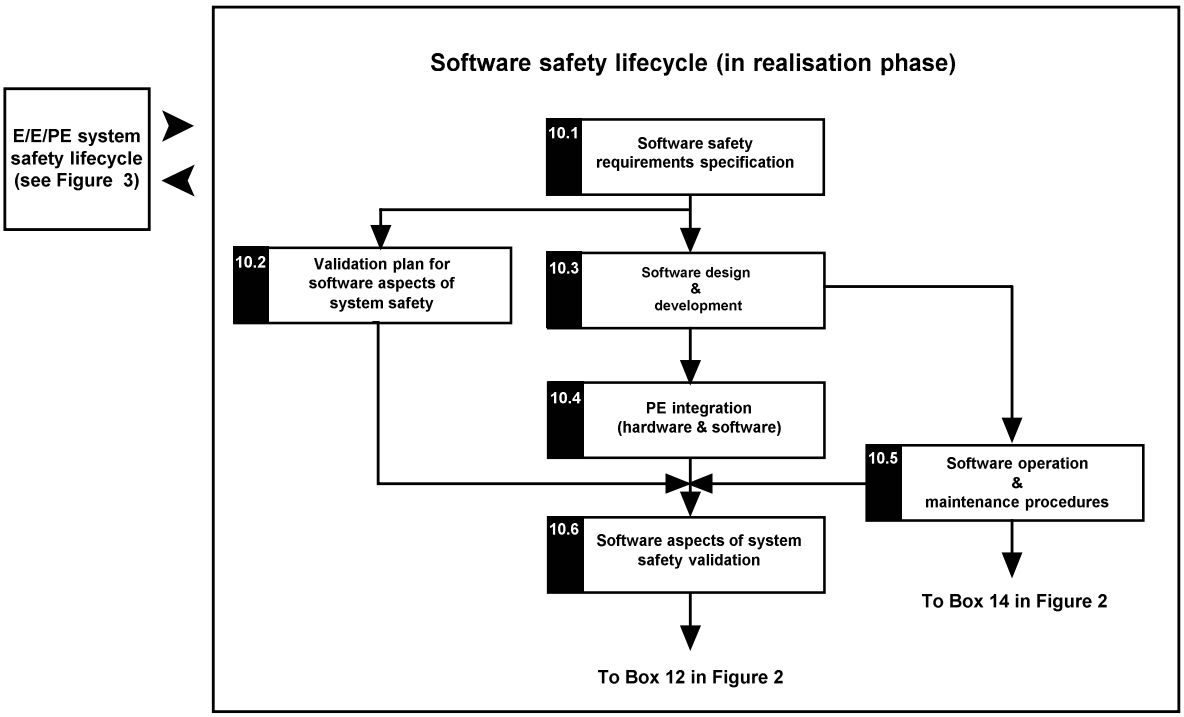


Figure 4 – Software safety lifecycle (in realisation phase)

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

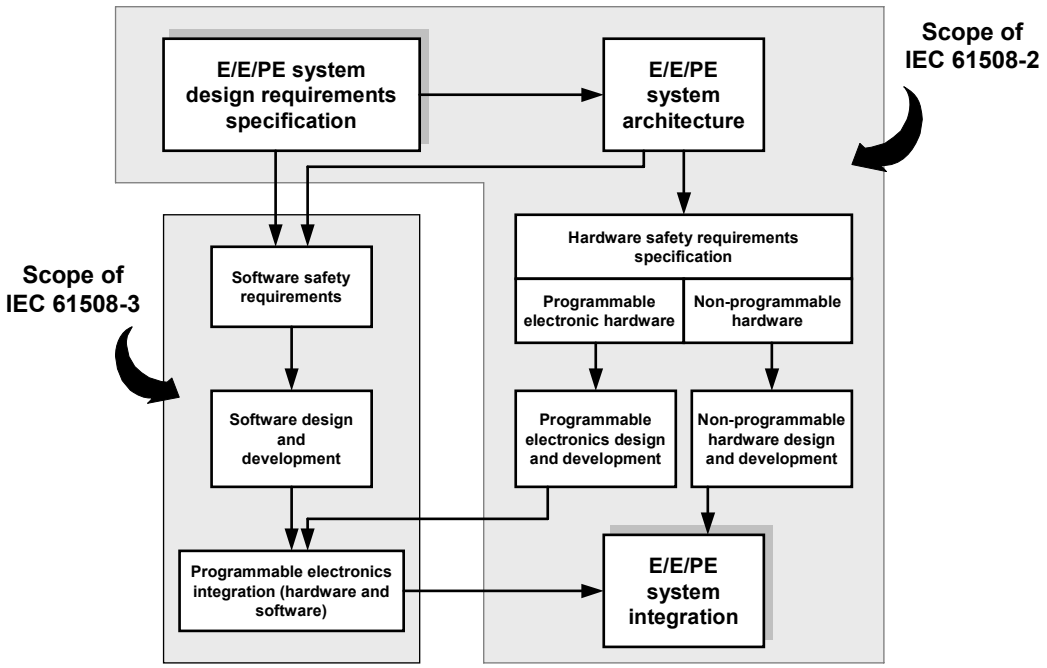


Figure 5 – Relationship and scope for IEC 61508-2 and IEC 61508-3

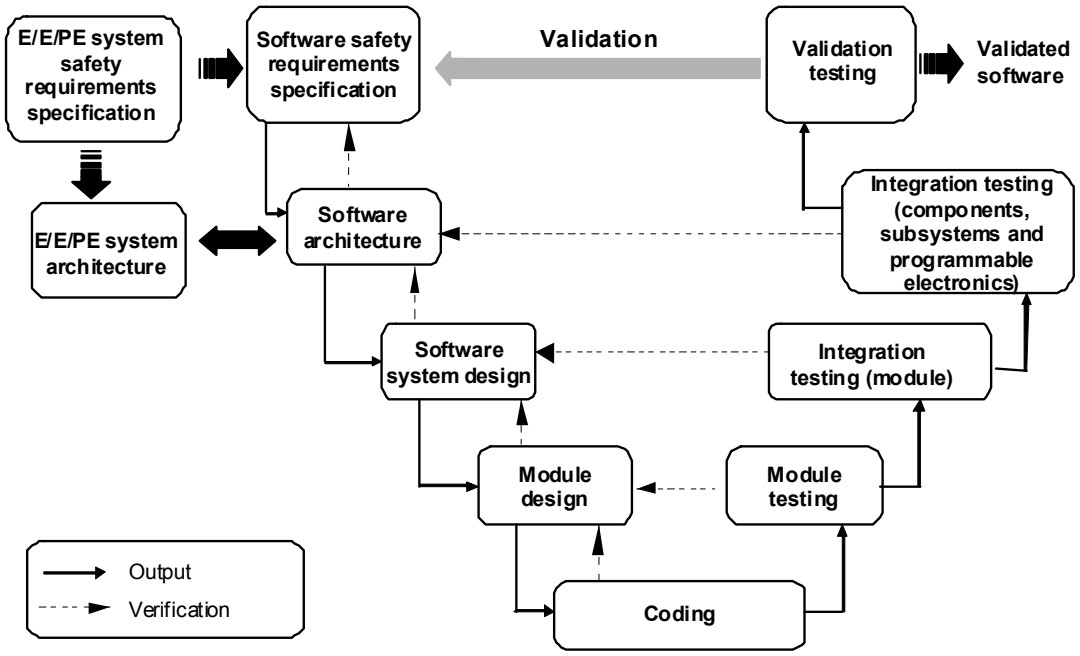


Figure 6 – Software systematic capability and the development lifecycle (the V-model)

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table 1 – Software safety lifecycle – overview

Safety lifecycle phase		Objectives	Scope	Require-ments subclause	Inputs (information required)	Outputs (information produced)
Figure 4 box number	Title					
10.1	Software safety requirements specification	To specify the requirements for safety-related software in terms of the requirements for software safety functions and the requirements for software systematic capability; To specify the requirements for the software safety functions for each E/E/PE safety-related system necessary to implement the required safety functions; To specify the requirements for software systematic capability for each E/E/PE safety-related system necessary to achieve the safety integrity level specified for each safety function allocated to that E/E/PE safety-related system	PE system; software system	7.2.2	E/E/PE safety requirements specification as developed during allocation (see IEC 61508-1) E/E/PE system safety requirements specification (from IEC 61508-2)	software safety requirements specification
10.2	Validation plan for software aspects of system safety	To develop a plan for validating the software aspects of system safety	PE system; software system	7.3.2	software safety requirements specification	validation plan for software aspects of system safety
10.3	Software design and development	Architecture: To create a software architecture that fulfils the specified requirements for safety-related software with respect to the required safety integrity level; To evaluate the requirements placed on the software by the hardware architecture of the E/E/PE safety-related system, including the significance of E/E/PE hardware/software interactions for safety of the equipment under control	PE system; software system	7.4.3	software safety requirements specification; E/E/PE system hardware architecture design (from IEC 61508-2)	software architecture design; software architecture integration test specification; software/ PE integration test specification (also required by IEC 61508-2)
10.3	Software design and development	Support tools and programming languages: To select a suitable set of tools, including languages and compilers, run-time system interfaces, user interfaces, and data formats and representations for the required safety integrity level, over the whole safety lifecycle of the software which assists verification, validation, assessment and modification	PE system; software system; support tools; programming language	7.4.4	software safety requirements specification; software architecture design	support tools and coding standards; selection of development tools

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table 1 *(continued)*

Safety lifecycle phase		Objectives	Scope	Require-ments subclause	Inputs (information required)	Outputs (information produced)
Figure 4 box number	Title					
10.3	Software design and development	Detailed design and development (software system design): To design and implement software that fulfils the specified requirements for safety-related software with respect to the required safety integrity level, which is analysable and verifiable, and which is capable of being safely modified	major elements and subsystems of software architectural design.	7.4.5	software architecture design; support tools and coding standards.	Software system design specification; software system integration test specification.
10.3	Software design and development	Detailed design and development (individual software module design): To design and implement software that fulfils the specified requirements for safety-related software with respect to the required safety integrity level, which is analysable and verifiable, and which is capable of being safely modified	software system design	7.4.5	software system design specification; support tools and coding standards	software module design specification; software module test specification
10.3	Software design and development	Detailed code implementation: To design and implement software that fulfils the specified requirements for safety-related software with respect to the required safety integrity level, which is analysable and verifiable, and which is capable of being safely modified	individual software modules	7.4.6	software module design specification; support tools and coding standards	source code listing; code review report
10.3	Software design and development	Software module testing: To verify that the requirements for safety-related software (in terms of the required software safety functions and the software systematic capability) have been achieved To show that each software module performs its intended function and does not perform unintended functions To ensure, in so far as it is appropriate, that configuration of PE systems by data fulfils the specified requirements for the software systematic capability	software modules	7.4.7	software module test specification; source code listing; code review report	software module test results; verified and tested software modules

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table 1 *(continued)*

Safety lifecycle phase		Objectives	Scope	Require-ments subclause	Inputs (information required)	Outputs (information produced)
Figure 4 box number	Title					
10.3	Software design and development	Software integration testing: To verify that the requirements for safety-related software (in terms of the required software safety functions and the software systematic capability) have been achieved To show that all software modules, elements and subsystems interact correctly to perform their intended function and do not perform unintended functions To ensure, in so far as it is appropriate, that configuration of PE systems by data fulfils the specified requirements for the software systematic capability	software architecture; software system	7.4.8	software system integration test specification	software system integration test results; verified and tested software system
10.4	Programmable electronics integration (hardware and software)	To integrate the software onto the target programmable electronic hardware; To combine the software and hardware in the safety-related programmable electronics to ensure their compatibility and to meet the requirements of the intended safety integrity level	program-mable electronics hardware; integrated software	7.5.2	software architecture integration test specification; software/PE integration test specification (also required by IEC 61508-2). Integrated programmable electronics	software architecture integration test results; programmable electronics integration test results; verified and tested integrated programmable electronics
10.5	Software operation and modification procedures	To provide information and procedures concerning software necessary to ensure that the functional safety of the E/E/PE safety-related system is maintained during operation and modification	as above	7.6.2	all above, as relevant	software operation and modification procedures
10.6	Software aspects of system safety validation	To ensure that the integrated system complies with the specified requirements for safety-related software at the intended safety integrity level	as above	7.7.2	validation plan for software aspects of system safety	software safety validation results; validated software
–	Software modification	To guide corrections, enhancements or adaptations to the validated software, ensuring that the required software systematic capability is sustained	as above	7.8.2	software modification procedures; software modification request	software modification impact analysis results; software modification log

--`,``,,`````````,,``,`,,``,`,`-`-`,,`,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table 1 *(continued)*

Safety lifecycle phase		Objectives	Scope	Require-ments subclause	Inputs (information required)	Outputs (information produced)
Figure 4 box number	Title					
–	Software verification	To test and evaluate the outputs from a given software safety lifecycle phase to ensure correctness and consistency with respect to the outputs and standards provided as input to that phase	depends on phase	7.9.2	appropriate verification plan (depends on phase)	appropriate verification report (depends on phase)
–	Software functional safety assessment	To investigate and arrive at a judgement on the software aspects of the functional safety achieved by the E/E/PE safety-related systems	all above phases	8	software functional safety assessment plan	software functional safety assessment report

7.2 Software safety requirements specification

NOTE This phase is Box 10.1 of Figure 4.

7.2.1 Objectives

7.2.1.1 The first objective of the requirements of this subclause is to specify the requirements for safety-related software in terms of the requirements for software safety functions and the requirements for software systematic capability.

7.2.1.2 The second objective of the requirements of this subclause is to specify the requirements for the software safety functions for each E/E/PE safety-related system necessary to implement the required safety functions.

7.2.1.3 The third objective of the requirements of this subclause is to specify the requirements for software systematic capability for each E/E/PE safety-related system necessary to achieve the safety integrity level specified for each safety function allocated to that E/E/PE safety-related system.

7.2.2 Requirements

NOTE 1 These requirements will in most cases be achieved by a combination of generic embedded software and application specific software. It is the combination of both that provides the features that satisfy the following subclauses. The exact division between generic and application specific software depends on the chosen software architecture (see 7.4.3).

NOTE 2 For the selection of appropriate techniques and measures (see Annexes A and B) to implement the requirements of this clause, the following properties (see Annex C for guidance on interpretation of properties, and Annex F of IEC 61508-7 for informal definitions) of the software safety requirements specification should be considered:

– completeness with respect to the safety needs to be addressed by software;

– correctness with respect to the safety needs to be addressed by software;

– freedom from intrinsic specification faults, including freedom from ambiguity;

– understandability of safety requirements;

– freedom from adverse interference of non-safety functions with the safety needs to be addressed by software;

– capability of providing a basis for verification and validation.

NOTE 3 The safety needs to be addressed by software is the set of safety functions and corresponding safety integrity requirements assigned to software functions by the design of the E/E/PE system. (The complete set of system safety needs is a larger set that includes also safety functions that do not depend on software). The completeness of the software safety requirements specification depends crucially on the effectiveness of earlier system lifecycle phases.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.2.2.1 If the requirements for safety-related software have already been specified for the E/E/PE safety-related system (see Clause 7 of IEC 61508-2), then the specification of software safety requirements need not be repeated.

7.2.2.2 The specification of the requirements for safety-related software shall be derived from the specified safety requirements of the E/E/PE safety-related system (see IEC 61508-2, 7), and any requirements of safety planning (see Clause 6). This information shall be made available to the software developer.

NOTE 1 This requirement does not mean that there will be no iteration between the developer of the E/E/PE system and the developer of the software (IEC 61508-2 and IEC 61508-3). As the safety-related software requirements and the software architecture become more precise, there may be an impact on the E/E/PE system hardware architecture, and for this reason close co-operation between the hardware and software developer is essential. See Figure 5.

NOTE 2 Where a software design incorporates pre-existing reusable software, that software may have been developed without taking account of the current system requirement specification. See 7.4.2.12 for the requirements on the pre-existing software to satisfy the software safety requirements specification.

7.2.2.3 The specification of the requirements for safety-related software shall be sufficiently detailed to allow the design and implementation to achieve the required safety integrity (including any requirement for independence, see 7.4.3 of IEC 61508-2), and to allow an assessment of functional safety to be carried out.

NOTE The level of detail of the specification may vary with the complexity of the application. An adequate specification of functional behaviour may include requirements for accuracy, timing and performance, capacity, robustness, overload tolerance, and other characterising properties of the specific application.

7.2.2.4 In order to address independence, a suitable common cause failure analysis shall be carried out. Where credible failure mechanisms are identified, effective defensive measures shall be taken.

NOTE See Annex F for techniques for achieving one aspect of independence of software.

7.2.2.5 The software developer shall evaluate the information in 7.2.2.2 to ensure that the requirements are adequately specified. In particular the software developer shall consider the following:

a) safety functions;

b) configuration or architecture of the system;

c) hardware safety integrity requirements (programmable electronics, sensors, and actuators);

d) software systematic capability requirements;

e) capacity and response time;

f) equipment and operator interfaces, including reasonably foreseeable misuse.

NOTE Compatibility with any applications already in existence should be considered.

7.2.2.6 If not already adequately defined in specified safety requirements of the E/E/PE safety-related system, all relevant modes of operation of the EUC, of the E/E/PE system, and of any equipment or system connected to the E/E/PE system shall be detailed in the specified requirements for safety-related software.

7.2.2.7 The software safety requirements specification shall specify and document any safety-related or relevant constraints between the hardware and the software.

7.2.2.8 To the extent required by the E/E/PE hardware architecture design, and considering the possible increase in complexity, the software safety requirements specification shall consider the following:

a) software self-monitoring (for examples see IEC 61508-7);

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

b) monitoring of the programmable electronics hardware, sensors, and actuators;

c) periodic testing of safety functions while the system is running;

d) enabling safety functions to be testable when the EUC is operational;

e) software functions to execute proof tests and all diagnostic tests in order to fulfil the safety integrity requirement of the E/E/PE safety-related system.

NOTE Increased complexity resulting from the above considerations may require the architecture to be revisited.

7.2.2.9 When the E/E/PE safety-related system is required to perform non-safety functions, then the specified requirements for safety-related software shall clearly identify the non-safety functions.

NOTE See 7.4.2.8 and 7.4.2.9 for requirements on non-interference between safety functions and non-safety functions.

7.2.2.10 The software safety requirements specification shall express the required safety properties of the product, but not of the project as this is covered by safety planning (see Clause 6 of 61508-1). With reference to 7.2.2.1 to 7.2.2.9, the following shall be specified as appropriate:

a) the requirements for the following software safety functions:
   1) functions that enable the EUC to achieve or maintain a safe state;
   2) functions related to the detection, annunciation and management of faults in the programmable electronics hardware;
   3) functions related to the detection, annunciation and management of sensor and actuators faults;
   4) functions related to the detection, annunciation and management of faults in the software itself (software self-monitoring);
   5) functions related to the periodic testing of safety functions on-line (i.e. in the intended operational environment);
   6) functions related to the periodic testing of safety functions off-line (i.e. in an environment where the EUC is not being relied upon for its safety function);
   7) functions that allow the PE system to be safely modified;
   8) interfaces to non safety-related functions;
   9) capacity and response time performance;
   10) interfaces between the software and the PE system;

       NOTE 1 They include both off-line and on-line programming facilities.

   11) safety-related communications (see 7.4.11 of IEC 61508-2).
b) the requirements for the software systematic capability:
   1) the safety integrity level(s) for each of the functions in a) above;

      NOTE 2 See Annex A of IEC 61508-5 for information concerning the allocation of safety integrity to software elements.

   2) independence requirements between functions.

7.2.2.11 Where software safety requirements are expressed or implemented by configuration data, the data shall be:

a) consistent with the system safety requirements;

b) expressed in terms of the permitted range and authorized combinations of its operational parameters;

c) defined in a manner which is compatible with the underlying software (for example sequence of execution, run time, data structures, etc.).

NOTE 1 This requirement on application data is particularly relevant to data-driven applications. These are characterized as follows: the source code is pre-existing and the primary objective of the development activity is to

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

provide assurance that the configuration data correctly states the behaviour required from the application. There may be complex dependencies between data items, and the validity of data may change over time.

NOTE 2 See Annex G for guidance on data-driven systems.

7.2.2.12 Where data defines the interface between software and external systems, the following performance characteristics shall be considered in addition to 7.4.11 of IEC 61508-2:

a) the need for consistency in terms of data definitions;

b) invalid, out of range or untimely values;

c) response time and throughput, including maximum loading conditions;

d) best case and worst case execution time, and deadlock;

e) overflow and underflow of data storage capacity.

7.2.2.13 Operational parameters shall be protected against:

a) invalid, out of range or untimely values;

b) unauthorized changes;

c) corruption.

NOTE 1 Protection against unauthorized changes should be considered, taking account of both software-based and non-software mechanisms. Note that effective protection against unauthorized software changes can have adverse effects on safety e.g. when changes are needed rapidly and in stressful conditions.

NOTE 2 Although a person can form part of a safety-related system (see Clause 1 of IEC 61508-1), human factor requirements related to the design of E/E/PE safety-related systems are not considered in detail in this standard. However, the following human considerations should be addressed where appropriate:

- An operator information system should use the pictorial layout and the terminology the operators are familiar with. It should be clear, understandable and free from unnecessary details and/or aspects;
- Information about the EUC displayed to the operator should follow closely the physical arrangement of the EUC;
- If several display contents to the operator are feasible and/or if the possible operator actions allow interactions whose consequences cannot be seen at one glance, the information displayed should automatically contain at each state of a display or an action sequence, which state of the sequence is reached, which operations are feasible and which possible consequences can be chosen.

7.3 Validation plan for software aspects of system safety

NOTE 1 This phase is Box 10.2 of Figure 4.

NOTE 2 Software usually cannot be validated separately from its underlying hardware and system environment.

7.3.1 Objective

The objective of the requirements of this subclause is to develop a plan for validating the safety-related software aspects of system safety.

7.3.2 Requirements

7.3.2.1 Planning shall be carried out to specify the steps, both procedural and technical, that will be used to demonstrate that the software satisfies its safety requirements.

7.3.2.2 The validation plan for software aspects of system safety shall consider the following:

a) details of when the validation shall take place;

b) details of those who shall carry out the validation;

c) identification of the relevant modes of the EUC operation including:

   1) preparation for use including setting and adjustment;

   2) start up, teach, automatic, manual, semi-automatic, steady state operation;

--`,,```,,,,````-`-`,,`,,`,`,,`---
Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

3) re-setting, shut down, maintenance;

4) reasonably foreseeable abnormal conditions and reasonably foreseeable operator misuse.

d) identification of the safety-related software which needs to be validated for each mode of EUC operation before commissioning commences;

e) the technical strategy for the validation (for example analytical methods, statistical tests etc.);

f) in accordance with item e), the measures (techniques) and procedures that shall be used for confirming that each safety function conforms with the specified requirements for the safety functions, and the specified requirements for software systematic capability;

g) the required environment in which the validation activities are to take place (for example, for tests this could include calibrated tools and equipment);

h) the pass/fail criteria;

i) the policies and procedures for evaluating the results of the validation, particularly failures.

NOTE These requirements are based on the general requirements given in 7.8 of IEC 61508-1.

7.3.2.3 The validation shall give a rationale for the chosen strategy. The technical strategy for the validation of safety-related software shall include the following information:

a) choice of manual or automated techniques or both;

b) choice of static or dynamic techniques or both;

c) choice of analytical or statistical techniques or both.

d) choice of acceptance criteria based on objective factors or expert judgment or both.

7.3.2.4 As part of the procedure for validating safety-related software aspects, the scope and contents of the validation plan for software aspects of system safety shall be agreed with the assessor or with a party representing the assessor, if required by the safety integrity level (see Clause 8 of IEC 61508-1). This procedure shall also make a statement concerning the presence of the assessor during testing.

7.3.2.5 The pass/fail criteria for accomplishing software validation shall include:

a) the required input signals with their sequences and their values;

b) the anticipated output signals with their sequences and their values; and

c) other acceptance criteria, for example memory usage, timing and value tolerances.

7.4 Software design and development

NOTE This phase is box 10.3 of Figure 4.

7.4.1 Objectives

7.4.1.1 The first objective of the requirements of this subclause is to create a software architecture that fulfils the specified requirements for safety-related software with respect to the required safety integrity level.

7.4.1.2 The second objective of the requirements of this subclause is to evaluate the requirements placed on the software by the hardware architecture of the E/E/PE safety-related system, including the significance of E/E/PE hardware/software interactions for safety of the equipment under control.

7.4.1.3 The third objective of the requirements of this subclause is to select a suitable set of tools, including languages and compilers, run-time system interfaces, user interfaces, and data formats and representations for the required safety integrity level, over the whole safety lifecycle of the software which assists verification, validation, assessment and modification.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.4.1.4 The fourth objective of the requirements of this subclause is to design and implement software that fulfils the specified requirements for safety-related software with respect to the required safety integrity level, which is analysable and verifiable, and which is capable of being safely modified.

7.4.1.5 The fifth objective of the requirements of this subclause is to verify that the requirements for safety-related software (in terms of the required software safety functions and the software systematic capability) have been achieved.

7.4.1.6 The sixth objective of the requirements of this subclause is to ensure, in so far as it is appropriate, that configuration of PE systems by data fulfils the specified requirements for the software systematic capability.

7.4.2 General requirements

7.4.2.1 Depending on the nature of the software development, responsibility for conformance with 7.4 can rest with the supplier of a safety related programming environment (e.g. PLC supplier) alone, or with the user of that environment (e.g. the application software developer) alone, or with both. The division of responsibility shall be determined during safety planning (see Clause 6).

NOTE See 7.4.3 for aspects of system and software architecture that are relevant to deciding on a practical division of responsibility.

7.4.2.2 In accordance with the required safety integrity level and the specific technical requirements of the safety function, the design method chosen shall possess features that facilitate:

a) abstraction, modularity and other features which control complexity;
b) the expression of:
    1) functionality;
    2) information flow between elements;
    3) sequencing and time related information;
    4) timing constraints;
    5) concurrency and synchronized access to shared resources;
    6) data structures and their properties;
    7) design assumptions and their dependencies;
    8) exception handling;
    9) design assumptions (pre-conditions, post-conditions, invariants);
    10) comments.
c) ability to represent several views of the design including structural and behavioural views;
d) comprehension by developers and others who need to understand the design;
e) verification and validation.

7.4.2.3 Testability and the capacity for safe modification shall be considered during the design activities in order to facilitate implementation of these properties in the final safety-related system.

NOTE Examples include maintenance modes in machinery and process plant.

7.4.2.4 The design method chosen shall possess features that facilitate software modification. Such features include modularity, information hiding and encapsulation.

NOTE See F.7.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.4.2.5 The design representations shall be based on a notation which is unambiguously defined or restricted to unambiguously defined features.

7.4.2.6 As far as practicable the design shall keep the safety-related part of the software simple.

7.4.2.7 The software design shall include, commensurate with the required safety integrity level, self-monitoring of control flow and data flow. On failure detection, appropriate actions shall be taken.

7.4.2.8 Where the software is to implement both safety and non-safety functions, then all of the software shall be treated as safety-related, unless adequate design measures ensure that the failures of non-safety functions cannot adversely affect safety functions.

7.4.2.9 Where the software is to implement safety functions of different safety integrity levels, then all of the software shall be treated as belonging to the highest safety integrity level, unless adequate independence between the safety functions of the different safety integrity levels can be shown in the design. It shall be demonstrated either (1) that independence is achieved by both in the spatial and temporal domains, or (2) that any violation of independence is controlled. The justification for independence shall be documented.

NOTE See Annex F for techniques for achieving one aspect of independence of software.

7.4.2.10 Where the systematic capability of a software element is lower than the safety integrity level of the safety function which the software element supports, the element shall be used in combination with other elements such that the systematic capability of the combination equals the safety integrity level of the safety function.

7.4.2.11 Where a safety function is implemented using a combination of software elements of known systematic capability, the systematic capability requirements of 7.4.3 of IEC 61508-2, shall apply to the combination of elements.

NOTE Distinguish consistently between (1) the *end-to-end safety function* that is supported by one or more elements and (2) the *element safety function* of each of the supporting elements. Where two elements combine to achieve a higher systematic capability in combination, each of the paired elements should be capable of preventing/mitigating the hazardous event, but the paired elements are not required to have identical element safety functions, and it is not required that each of the paired elements is independently capable of providing the whole safety functionality demanded from the combination.

EXAMPLE An electronic engine throttle control where the *end-to-end safety function* is “prevent undemanded acceleration”. The *end-to-end safety function* is implemented by two processors. The *element safety function* of the primary controller is the ideal demand/response behaviour of the throttle. The *element safety function* of the secondary processor is a diverse monitor (see IEC 61508-7 C.3.4) and applies an emergency stop if necessary. The combination of the two processors gives higher confidence that the end-to-end safety function “prevent undemanded acceleration” will be achieved.

7.4.2.12 Where a pre-existing software element is reused to implement all or part of a safety function, the element shall meet both requirements a) and b) below for systematic safety integrity:

a) meet the requirements of one of the following compliance routes:
   - Route 1_S: compliant development. Compliance with the requirements of this standard for the avoidance and control of systematic faults in software;
   - Route 2_S: proven in use. Provide evidence that the element is proven in use. See 7.4.10 of IEC 61508-2;
   - Route 3_S:assessment of non-compliant development. Compliance with 7.4.2.13.

NOTE 1 Route 1_S, 2_S and 3_S are the element compliance routes of 7.4.2.2 c) of IEC 61508-2 with particular reference to software elements. They are reproduced here for convenience only, and to minimize references back to IEC 61508-2.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

NOTE 2 See 3.2.8 of IEC 61508-4. The pre-existing software could be a commercially available product, or it could have been developed by some organisation for a previous product or system. Pre-existing software may or may not have been developed in accordance with the requirements of this standard.

NOTE 3 Requirements on pre-existing elements apply to a run-time library or an interpreter.

b) provide a safety manual (see Annex D of IEC 61508-2 and Annex D of this standard) that gives a sufficiently precise and complete description of the element to make possible an assessment of the integrity of a specific safety function that depends wholly or partly on the pre-existing software element.

NOTE 4 The safety manual may be derived from the element supplier's own documentation and records of the element supplier's development process, or may be created or supplemented by additional qualification activities undertaken by the developer of the safety related system or by third parties. In some cases, reverse engineering may be required to create specification or design documentation adequate to meet the requirements of this clause, subject to the prevailing legal conditions (e.g. copyright or intellectual property rights).

NOTE 5 The justification of the element may be developed during safety planning (see Clause 6).

7.4.2.13 To comply with Route 3_s a pre-existing software element shall meet all of the following requirements a) to i):

a) The software safety requirements specification for the element in its new application shall be documented to the same degree of precision as would be required by this standard for any safety related element of the same systematic capability. The software safety requirements specification shall cover the functional and safety behaviour as applicable to the element in its new application and as specified in 7.2. See Table A.1.

b) The justification for use of a software element shall provide evidence that the desirable safety properties specified in the referenced subclauses (i.e. 7.2.2, 7.4.3, 7.4.4, 7.4.5, 7.4.6, 7.4.7, 7.5.2, 7.7.2, 7.8.2, 7.9.2, and Clause 8) have been considered, taking account of the guidance in Annex C.

c) The element's design shall be documented to a degree of precision, sufficient to provide evidence of compliance with the requirement specification and the required systematic capability. See 7.4.3, 7.4.5 and 7.4.6, and Tables A.2 and A.4 of Annex A.

d) The evidence required in 7.4.2.13 a) and 7.4.2.13 b) shall cover the software's integration with the hardware. See 7.5 and Table A.6 of Annex A.

e) There shall be evidence that the element has been subject to verification and validation using a systematic approach with documented testing and review of all parts of the element's design and code. See 7.4.7, 7.4.8, 7.5, 7.7 and 7.9 and Tables A.5 to A.7 and A.9 of Annex A as well as related tables in Annex B.

NOTE 1 Positive operational experience may be used to satisfy black-box and probabilistic testing requirements [see Tables A.7 and B.3].

f) Where the software element provides functions which are are not required in the safety related system, then evidence shall be provided that the unwanted functions will not prevent the E/E/PE system from meeting its safety requirements.

NOTE 2 Ways to meet this requirement include:

- removing the functions from the build;
- disabling the functions;
- appropriate system architecture (e.g. partitioning, wrappers, diversity, checking the credibility of outputs);
- extensive testing.

g) There shall be evidence that all credible failure mechanisms of the software element have been identified and that appropriate mitigation measures have been implemented.

NOTE 3 Appropriate mitigation measures include:

- appropriate system architecture (e.g. partitioning, wrappers, diversity, credibility of checking of outputs);
- exception handling.

h) The planning for use of the element shall identify the configuration of the software element, the software and hardware run-time environment and if necessary the configuration of the compilation / linking system.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

i) The justification for use of the element shall be valid for only those applications which respect the assumptions made in the compliant item safety manual for the element (see Annex D of IEC 61508-2 and Annex D).

7.4.2.14 This Subclause 7.4.2 shall, in so far as it is appropriate, apply to data and data generation languages.

NOTE See Annex G for guidance on data-driven systems.

a) Where a PE system consists of pre-existing functionality that is configured by data to meet specific application requirements, the design of the application software shall be commensurate with the degree of application configurability, pre-delivered existing functionality and complexity of the PE safety-related system.

b) Where the safety-related functionality of a PE system is determined significantly or predominantly by configuration data, appropriate techniques and measures shall be used to prevent the introduction of faults during the design, production, loading and modification of the configuration data and to ensure that the configuration data correctly states the application logic.

c) The specification of data structures shall be:

   1) consistent with the functional requirements of the system, including the application data;

   2) complete;

   3) self consistent;

   4) such that the data structures are protected against alteration or corruption.

d) Where a PE System consists of pre-existing functionality that is configured by data to meet specific application requirements, the configuration process itself shall be documented appropriately.

7.4.3 Requirements for software architecture design

NOTE 1 The software architecture defines the major elements and subsystems of the software, how they are interconnected, and how the required attributes, particularly safety integrity, will be achieved. It also defines the overall behaviour of the software, and how software elements interface and interact. Examples of major software elements include operating systems, databases, EUC input/output subsystems, communication subsystems, application program(s), programming and diagnostic tools, etc.

NOTE 2 In certain industrial sectors the software architecture would be called a function description or functional design specification (although these documents could also include the hardware).

NOTE 3 In some contexts of user application programming, particularly in PLCs (see Annex E of IEC 61508-6), the software architecture is provided by the supplier as a standard feature of the product. The supplier would, under this standard, be required to assure the user of the compliance of his products to the requirements of 7.4. The user tailors the PLC to the application by using the standard programming facilities, for example ladder logic. The requirements of 7.4.3 to 7.4.8 still apply. The requirement to define and document the software architecture can be seen as information that the user would use to select the PLC (or equivalent) for the application.

NOTE 4 From a safety viewpoint, the software architecture phase is where the basic safety strategy is developed for the software.

NOTE 5 Although the IEC 61508 series sets numerical target failure measures for safety functions carried out by E/E/PE safety-related systems, systematic safety integrity is usually unquantified (see 3.5.6 of IEC 61508-4), and software safety integrity (see 3.5.5 of IEC 61508-4) is defined as a systematic capability on a confidence scale of 1-4 (see 3.5.9 of IEC 61508-4). This standard recognizes that a software failure can be safe or unsafe depending on the specific use of the software The system/software architecture needs to be such that unsafe failures of an element are limited by some architectural constraint, and that development methods should take account of these constraints. This standard applies development and validation techniques with rigour that is qualitatively consistent with the required systematic capability.

NOTE 6 For the selection of appropriate techniques and measures (see Annexes A and B) to implement the requirements of this clause, the following properties (see Annex C for guidance on interpretation of properties, and Annex F of IEC 61508-7 for informal definitions) of the software architecture design should be considered:

– completeness with respect to software safety requirements specification;

– correctness with respect to software safety requirements specification;

– freedom from intrinsic design faults;

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

– simplicity and understandability;

– predictability of behaviour;

– verifiable and testable design;

– fault tolerance;

– defence against common cause failure from external events.

7.4.3.1 Depending on the nature of the software development, responsibility for conformance with 7.4.4 can rest with multiple parties. The division of responsibility shall be documented during safety planning (see Clause 6 of IEC 61508-1).

7.4.3.2 The software architecture design shall be established by the software supplier and/or developer, and shall be detailed. The software architecture design shall:

a) select and justify (see 7.1.2.7) an integrated set of techniques and measures necessary during the software safety lifecycle phases to satisfy the software safety requirements specification at the required safety integrity level. These techniques and measures include software design strategies for both fault tolerance (consistent with the hardware) and fault avoidance, including (where appropriate) redundancy and diversity;

b) be based on a partitioning into elements/subsystems, for each of which the following information shall be provided:

   1) whether the elements/subsystems have been previously verified, and if yes, their verification conditions;

   2) whether each subsystem/element is safety-related or not;

   3) software systematic capability of the subsystem/element.

c) determine all software/hardware interactions and evaluate and detail their significance;

NOTE Were the software/hardware interaction is already determined by the system architecture, it is sufficient to refer to the system architecture.

d) use a notation to represent the architecture which is unambiguously defined or restricted to unambiguously defined features;

e) select the design features to be used for maintaining the safety integrity of all data. Such data may include plant input-output data, communications data, operator interface data, maintenance data and internal database data;

f) specify appropriate software architecture integration tests to ensure that the software architecture satisfies the software safety requirements specification at the required safety integrity level.

7.4.3.3 Any changes required to the E/E/PE System Safety Requirements Specification (see 7.2.2) after applying 7.4.3.2 shall be agreed with the E/E/PE developer and documented.

NOTE There will inevitably be iteration between the hardware and software architecture (see Figure 5) and there is therefore a need to discuss with the hardware developer such issues as the test specification for the integration of the programmable electronics hardware and the software (see 7.5).

7.4.4 Requirements for support tools, including programming languages

NOTE For the selection of appropriate techniques and measures (see Annexes A and B) to implement the requirements of this clause, the following properties (see Annex C for guidance on interpretation of properties, and Annex F of IEC 61508-7 for informal definitions) of support tools should be considered:

– the degree to which the tool supports the production of software with the required software properties;

– the clarity of the operation and functionality of the tool;

– the correctness and repeatability of the output.

7.4.4.1 A software on-line support tool shall be considered to be a software element of the safety-related system

NOTE See 3.2.10 and 3.2.11 of IEC 61508-4 for examples of on-line and off-line tools.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.4.4.2 Software off-line support tools shall be selected as a coherent part of the software development activities.

NOTE 1 See 7.1.2 for software development lifecycle requirements.

NOTE 2 Appropriate off-line tools to support the development of software should be used in order to increase the integrity of the software by reducing the likelihood of introducing or not detecting faults during the development. Examples of tools relevant to the phases of the software development lifecycle include:

a) transformation or translation tools that convert a software or design representation (e.g. text or a diagram) from one abstraction level to another level: design refinement tools, compilers, assemblers, linkers, binders, loaders and code generation tools;

b) verification and validation tools such as static code analysers, test coverage monitors, theorem proving assistants, and simulators;

c) diagnostic tools used to maintain and monitor the software under operating conditions;

d) infrastructure tools such as development support systems;

e) configuration control tools such as version control tools;

f) application data tools that produce or maintain data which are required to define parameters and to instantiate system functions. Such data includes function parameters, instrument ranges, alarm and trip levels, output states to be adopted at failure, geographical layout.

NOTE 3 Off-line support tools should be selected to be integrated. In this context, tools are integrated if they work co-operatively such that the outputs from one tool have suitable content and format for automatic input to a subsequent tool, thus minimising the possibility of introducing human error in the reworking of intermediate results.

NOTE 4 Off-line support tools should be selected to be compatible with the needs of the application, of the safety related system, and of the integrated toolset.

NOTE 5 The availability of suitable tools to supply the services that are necessary over the whole lifetime of the E/E/PE safety-related system (e.g. tools to support specification, design, implementation, documentation, modification) should be considered.

NOTE 6 Consideration should be given to the competence of the users of the selected tools. See Clause 6 of IEC 61508-1 for competence requirements.

7.4.4.3 The selection of the off-line support tools shall be justified.

7.4.4.4 All off-line support tools in classes T2 and T3 shall have a specification or product documentation which clearly defines the behaviour of the tool and any instructions or constraints on its use. See 7.1.2 for software development lifecycle requirements, and 3.2.11 of IEC 61508-4 for categories of software off-line support tool.

NOTE This “specification or product documentation” is not a safety manual for compliant items (see Annex D of 61508-2 and also of this standard) for the tool itself. The safety manual for compliant item relates to a pre-existing element that is incorporated into the executable safety related system. Where a pre-existing element has been generated by a T3 tool and then incorporated into the executable safety related system, then any relevant information (e.g. the documentation for an optimising compiler may indicate that the evaluation order of function parameters is not guaranteed) from the tool’s “specification or product documentation” should be included in the compliant item safety manual that makes possible an assessment of the integrity of a specific safety function that depends wholly or partly on the incorporated element.”

7.4.4.5 An assessment shall be carried out for offline support tools in classes T2 and T3 to determine the level of reliance placed on the tools, and the potential failure mechanisms of the tools that may affect the executable software. Where such failure mechanisms are identified, appropriate mitigation measures shall be taken.

NOTE 1 Software HAZOP is one technique to analyse the consequences of potential software tool failures.

NOTE 2 Examples of mitigation measures include: avoiding known bugs, restricted use of the tool functionality, checking the tool output, use of diverse tools for the same purpose.

7.4.4.6 For each tool in class T3, evidence shall be available that the tool conforms to its specification or documentation. Evidence may be based on a suitable combination of history of successful use in similar environments and for similar applications (within the organisation or other organisations), and of tool validation as specified in 7.4.4.7.

NOTE 1 A version history may provide assurance of maturity of the tool, and a record of the errors / ambiguities that should be taken into account when the tool is used in the new development environment.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

NOTE 2 The evidence listed for T3 may also be used for T2 tools in judging the correctness of their results.

7.4.4.7 The results of tool validation shall be documented covering the following results:

a) a chronological record of the validation activities;

b) the version of the tool product manual being used;

c) the tool functions being validated;

d) tools and equipment used;

e) the results of the validation activity; the documented results of validation shall state either that the software has passed the validation or the reasons for its failure;

f) test cases and their results for subsequent analysis;

g) discrepancies between expected and actual results.

7.4.4.8 Where the conformance evidence of 7.4.4.6 is unavailable, there shall be effective measures to control failures of the executable safety related system that result from faults that are attributable to the tool.

NOTE An example of a measure would be the generation of diverse redundant code which allows the detection and control of failures of the executable safety related system as a result of faults that have been introduced into the executable safety related system by a translator.

7.4.4.9 The compatibility of the tools of an integrated toolset shall be verified.

Note: tools are integrated if they work co-operatively such that the outputs from one tool have suitable content and format for automatic input to a subsequent tool, thus minimizing the possibility of introducing human error in the reworking of intermediate results. See IEC 61508-7 B.3.5.

7.4.4.10 To the extent required by the safety integrity level, the software or design representation (including a programming language) selected shall:

a) have a translator which has been assessed for fitness for purpose including, where appropriate, assessment against the international or national standards;

b) use only defined language features;

c) match the characteristics of the application;

d) contain features that facilitate the detection of design or programming mistakes;

e) support features that match the design method.

NOTE 1 A programming language is a class of software or design representations. A translator converts a software or design representation (e.g. text or a diagram) from one abstraction level to another level. Examples of translators include: design refinement tools, compilers, assemblers, linkers, binders, loaders and code generation tools.

NOTE 2 The assessment of a translator may be performed for a specific application project, or for a class of applications. In the latter case all necessary information on the tool (the “specification or product manual”, see 7.4.4.4) regarding the intended and appropriate use of the tool should be available to the user of the tool. The assessment of the tool for a specific project may then be reduced to checking general suitability of the tool for the project and compliance with the “specification or product manual” (i.e. proper use of the tool). Proper use might include additional verification activities within the specific project.

NOTE 3 A validation suite (i.e. a set of test programs whose correct translation is known in advance) may be used to evaluate the fitness for purpose of a translator according to defined criteria, which should include functional and non-functional requirements. For the functional translator requirements, dynamic testing may be a main validation technique. If possible an automatic testing suite should be used.

7.4.4.11 Where 7.4.4.10 cannot be fully satisfied, the fitness for purpose of the language, and any additional measures which address any identified shortcomings of the language shall be justified.

7.4.4.12 Programming languages for the development of all safety-related software shall be used according to a suitable programming language coding standard.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

NOTE See IEC 61508-7 for guidance on coding standard aspects that relate to software safety.

7.4.4.13 A programming language coding standard shall specify good programming practice, proscribe unsafe language features (for example, undefined language features, unstructured designs, etc.), promote code understandability, facilitate verification and testing, and specify procedures for source code documentation. Where practicable, the following information shall be contained in the source code:

a) legal entity (for example company, author(s), etc.);

b) description;

c) inputs and outputs;

d) configuration management history.

7.4.4.14 Where automatic code generation or similar automatic translation takes place, the suitability of the automatic translator for safety-related system development shall be assessed at the point in the development lifecycle where development support tools are selected.

7.4.4.15 Where off-line support tools of classes T2 and T3 generate items in the configuration baseline, configuration management shall ensure that information on the tools is recorded in the configuration baseline. This includes in particular:

a) the identification of the tool and its version;

b) the identification of the configuration baseline items for which the tool version has been used;

c) the way the tool was used (including the tool parameters, options and scripts selected) for each configuration baseline item.

NOTE The objective of this clause is to allow the baseline to be reconstructed.

7.4.4.16 Configuration management shall ensure that for tools in classes T2 and T3, only qualified versions are used.

7.4.4.17 Configuration management shall ensure that only tools compatible with each other and with the safety-related system are used.

NOTE The safety-related system hardware may also impose compatibility constraints on software tools e.g. a processor emulator needs to be an accurate model of the real processor electronics.

7.4.4.18 Each new version of off-line support tool shall be qualified. This qualification may rely on evidence provided for an earlier version if sufficient evidence is provided that:

a) the functional differences (if any) will not affect tool compatibility with the rest of the toolset; and

b) the new version is unlikely to contain significant new, unknown faults.

NOTE Evidence that the new version is unlikely to contain significant new, unknown faults may be based on (1) a clear identification of the changes made, (2) an analysis of the verification and validation actions performed on the new version, and (3) any existing operational experience from other users that is relevant to the new version.

7.4.4.19 Depending on the nature of the software development, responsibility for conformance with 7.4.4 can rest with multiple parties. The division of responsibility shall be documented during safety planning (see Clause 6 of IEC 61508-1).

7.4.5 Requirements for detailed design and development – software system design

NOTE 1 Detailed design is defined here to mean software system design: the partitioning of the major elements in the architecture into a system of software modules; individual software module design; and coding. In small applications, software system design and architectural design may be combined.

NOTE 2 The nature of detailed design and development will vary with the nature of the software development activities and the software architecture (see 7.4.3). In some contexts of application programming, for example ladder logic and function blocks, detailed design can be considered as configuring rather than programming.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

However it is still good practice to design the software in a structured way, including organising the software into a modular structure that separates out (as far as possible) safety-related parts; including range checking and other features that provide protection against data input mistakes; using previously verified software modules; and providing a design that facilitates future software modifications.

NOTE 3 For the selection of appropriate techniques and measures (see Annexes A and B) to implement the requirements of this clause, the following properties (see Annex C for guidance on interpretation of properties, and Annex F of IEC 61508-7 for informal definitions) of the design and development should be considered:

– completeness with respect to software safety requirements specification;

– correctness with respect to software safety requirements specification;

– freedom from intrinsic design faults;

– simplicity and understandability

– predictability of behaviour;

– verifiable and testable design;

– fault tolerance / fault detection;

– freedom from common cause failure.

7.4.5.1 Depending on the nature of the software development, responsibility for conformance with 7.4.5 can rest with multiple parties. The division of responsibility shall be documented during safety planning (see Clause 6 of IEC 61508-1).

7.4.5.2 The following information shall be available prior to the start of detailed design: the specification of requirements for the E/E/PE safety related system; the software architecture design; the validation plan for software aspects of system safety.

7.4.5.3 The software shall be produced to achieve modularity, testability, and the capability for safe modification.

7.4.5.4 For each major element/subsystem in the software architecture design, further refinement of the design shall be based on a partitioning into software modules (i.e. the specification of the software system design). The design of each software module and the verification to be applied to each software module shall be specified.

NOTE 1 For pre-existing software elements, see 7.4.2.

NOTE 2 Verification includes testing and analysis.

7.4.5.5 Appropriate software system integration tests shall be specified to ensure that the software system satisfies the software safety requirements specification at the required safety integrity level.

7.4.6 Requirements for code implementation

NOTE To the extent required by the safety integrity level, the source code shall possess the following properties (see Annexes A and B for specific techniques, and see Annex C for guidance on interpretation of properties) of code should be considered:

- be readable, understandable and testable;
- satisfy the specified requirements for software module design (see 7.4.5);
- satisfy the specified requirements of the coding standards (see 7.4.4);
- satisfy all relevant requirements specified during safety planning (see Clause 6).

7.4.6.1 Each module of software code shall be reviewed. Where the code is produced by an automatic tool, the requirements of 7.4.4 shall be met. Where the source code consists of reused pre-existing software, the requirements of 7.4.2 shall be met.

NOTE Code review is a verification activity (see 7.9). Code review can be carried out by means of an inspection of the code: (1) by an individual; (2) by a software walk-though (see IEC 61508-7 C.5.15); or (3) by a formal inspection (see IEC 61508-7 C.5.14), in increasing order of rigour.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.4.7 Requirements for software module testing

NOTE 1 Testing that the software module correctly satisfies its test specification is a verification activity (see 7.9). It is the combination of code review and software module testing that provides assurance that a software module satisfies its associated specification, i.e. it is verified.

NOTE 2 For the selection of appropriate techniques and measures (see Annexes A and B) to implement the requirements of this clause, the following properties (see Annex C for guidance on interpretation of properties, and Annex F of IEC 61508-7 for informal definitions) of the software module testing should be considered:

– completeness of testing with respect to the software design specification;

– correctness of testing with respect to the software design specification (successful completion);

– repeatability;

– precisely defined testing configuration.

7.4.7.1 Each software module shall be verified as required by the software module test specification that was developed during software system design (see 7.4.5).

NOTE Verification includes testing and analysis.

7.4.7.2 This verification shall show whether or not each software module performs its intended function and does not perform unintended functions.

NOTE 1 This does not imply testing of all input combinations, nor of all output combinations. Testing all equivalence classes or structure based testing may be sufficient. Boundary value analysis or control flow analysis may reduce the test cases to an acceptable number. Analysable programs make the requirements easier to fulfil. See Annex C of IEC 61508-7 for these techniques.

NOTE 2 Where the development uses formal methods, formal proofs or assertions, such tests may be reduced in scope. See Annex C of IEC 61508-7 for these techniques.

NOTE 3 Although systematic safety integrity is usually unquantified (see 3.5.6 of IEC 61508-4), quantified statistical evidence (e.g. statistical testing, reliability growth) is acceptable if all the relevant conditions for statistically valid evidence are satisfied e.g. see Annex D of IEC 61508-7.

NOTE 4 If the module is simple enough to make practicable an exhaustive test, then this can be the most efficient way to demonstrate conformance.

7.4.7.3 The results of the software module testing shall be documented.

7.4.7.4 The procedures for corrective action on not passing the test shall be specified.

7.4.8 Requirements for software integration testing

NOTE Testing that the software is correctly integrated is a verification activity (see 7.9).

7.4.8.1 Software integration tests shall be specified during the design and development phase (see 7.4.5).

7.4.8.2 The software system integration test specification shall state the following:

a) the division of the software into manageable integration sets;

b) test cases and test data;

c) types of tests to be performed;

d) test environment, tools, configuration and programs;

e) test criteria on which the completion of the test will be judged;

f) procedures for corrective action on failure of test.

7.4.8.3 The software shall be tested in accordance with the software integration tests specified in the software system integration test specification. These tests shall show that all software modules and software elements/subsystems interact correctly to perform their intended function and do not perform unintended functions.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

NOTE 1 This does not imply testing of all input combinations, nor of all output combinations. Testing all equivalence classes or structure based testing may be sufficient. Boundary value analysis or control flow analysis may reduce the test cases to an acceptable number. Analysable programs make the requirements easier to fulfil. See Annex C of IEC 61508-7 for these techniques.

NOTE 2 Where the development uses formal methods, formal proofs or assertions, such tests may be reduced in scope. See Annex C of IEC 61508-7 for these techniques.

NOTE 3 Although systematic safety integrity is usually unquantified (see 3.5.6 of IEC 61508-4), quantified statistical evidence (e.g. statistical testing, reliability growth) is acceptable if all the relevant conditions for statistically valid evidence are satisfied e.g. see Annex D of IEC 61508-7.

7.4.8.4 The results of software integration testing shall be documented, stating the test results, and whether the objectives and the test criteria have been met. If there is a failed integration test, the reasons for the failure shall be documented.

7.4.8.5 During software integration, any modification to the software shall be subject to an impact analysis which shall determine all software modules impacted, and the necessary re-verification and re-design activities.

7.5 Programmable electronics integration (hardware and software)

NOTE This phase is box 10.4 of Figure 4.

7.5.1 Objectives

7.5.1.1 The first objective of the requirements of this subclause is to integrate the software onto the target programmable electronic hardware.

7.5.1.2 The second objective of the requirements of this subclause is to combine the software and hardware in the safety-related programmable electronics to ensure their compatibility and to meet the requirements of the intended safety integrity level.

NOTE 1 Testing that the software is correctly integrated with the programmable electronic hardware is a verification activity (see 7.9).

NOTE 2 Depending on the nature of the application, these activities may be combined with 7.4.8.

7.5.2 Requirements

NOTE For the selection of appropriate techniques and measures (see Annexes A and B) to implement the requirements of this clause, the following properties (see Annex C for guidance on interpretation of properties, and Annex F of IEC 61508-7 for informal definitions) of the integration should be considered:

– completeness of integration with respect to the design specifications;

– correctness of integration with respect to the design specifications (successful completion);

– repeatability;

– precisely defined integration configuration.

7.5.2.1 Integration tests shall be specified during the design and development phase (see 7.4.3) to ensure the compatibility of the hardware and software in the safety-related programmable electronics.

NOTE Close co-operation with the developer of the E/E/PE system may be required in order to develop the integration tests.

7.5.2.2 The software/PE integration test specification (hardware and software) shall state the following:

a) the split of the system into integration levels;

b) test cases and test data;

c) types of tests to be performed;

d) test environment including tools, support software and configuration description;

e) test criteria on which the completion of the test will be judged.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.5.2.3 The software/PE integration test specification (hardware and software) shall distinguish between those activities which can be carried out by the developer on his premises and those that require access to the user's site.

7.5.2.4 The software/PE integration test specification (hardware and software) shall distinguish between the following activities:

a) merging of the software system on to the target programmable electronic hardware;

b) E/E/PE integration, i.e. adding interfaces such as sensors and actuators;

c) applying the E/E/PE safety-related system to the EUC.

NOTE Items b) and c) are covered by IEC 61508-1 and IEC 61508-2 and are included here to put item a) in context and for completeness. They are not normally the responsibility of the software developers.

7.5.2.5 The software shall be integrated with the safety-related programmable electronic hardware in accordance with the software/PE integration test specification (hardware and software).

7.5.2.6 During the integration testing of the safety-related programmable electronics (hardware and software), any change to the integrated system shall be subject to an impact analysis. The impact analysis shall determine all software modules impacted, and the necessary re-verification activities.

7.5.2.7 Test cases and their expected results shall be documented for subsequent analysis.

7.5.2.8 The integration testing of the safety-related programmable electronics (hardware and software) shall be documented, stating the test results, and whether the objectives and the test criteria have been met. If there is a failure, the reasons for the failure shall be documented. Any resulting modification or change to the software shall be subject to an impact analysis which shall determine all software elements/modules impacted, and the necessary re-verification and re-design activities.

7.6 Software operation and modification procedures

NOTE This phase is box 10.5 of Figure 4.

7.6.1 Objective

The objective of the requirements of this subclause is to provide information and procedures concerning software necessary to ensure that the functional safety of the E/E/PE safety-related system is maintained during operation and modification.

7.6.2 Requirements

The requirements are given in 7.6 of IEC 61508-2 and in 7.8 of this standard.

NOTE In this standard software (unlike hardware) is not capable of being maintained: it is always modified.

7.7 Software aspects of system safety validation

NOTE 1 This phase is box 10.6 of Figure 4.

NOTE 2 Software usually cannot be validated separately from its underlying hardware and system environment.

7.7.1 Objective

The objective of the requirements of this subclause is to ensure that the integrated system complies with the software safety requirements specification at the required safety integrity level.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.7.2 Requirements

NOTE For the selection of appropriate techniques and measures (see Annexes A and B) to implement the requirements of this clause, the following properties (see Annex C for guidance on interpretation of properties, and Annex F of IEC 61508-7 for informal definitions) of safety validation should be considered:

– completeness of validation with respect to the software design specification;

– correctness of validation with respect to the software design specification (successful completion);

– repeatability;

– precisely defined validation configuration.

7.7.2.1 If the compliance with the requirements for safety-related software has already been established in the safety validation planning for the E/E/PE safety-related system (see 7.7 of IEC 61508-2), then the validation need not be repeated.

7.7.2.2 The validation activities shall be carried out as specified the in validation plan for software aspects of system safety.

7.7.2.3 Depending on the nature of the software development, responsibility for conformance with 7.7 can rest with multiple parties. The division of responsibility shall be documented during safety planning (see Clause 6 of IEC 61508-1).

7.7.2.4 The results of validating the software aspects of system safety shall be documented.

7.7.2.5 For each safety function, software safety validation shall document the following results:

a) a chronological record of the validation activities that will permit the sequence of activities to be retraced;

   NOTE When recording test results, it is important to be able to retrace the sequence of activities. The emphasis of this requirement is on retracing a sequence of activities, and not on producing a timed/dated list of documents.

b) the version of the validation plan for software aspects of system safety (see 7.3) being used;

c) the safety function being validated (by test or analysis), together with reference to the validation plan for software aspects of system safety;

d) tools and equipment used together with calibration data;

e) the results of the validation activity;

f) discrepancies between expected and actual results.

7.7.2.6 When discrepancies occur between expected and actual results, the analysis made and the decisions taken on whether to continue the validation, or to issue a change request and return to an earlier part of the development lifecycle, shall be documented as part of the results of validating the software aspects of system safety.

NOTE The requirements of 7.7.2.2 to 7.7.2.6 are based on the general requirements given in 7.14 of IEC 61508-1.

7.7.2.7 The validation of safety-related software aspects of system safety shall meet the following requirements:

a) testing shall be the main validation method for software; analysis, animation and modelling may be used to supplement the validation activities;

b) the software shall be exercised by simulation of:

   1) input signals present during normal operation;

   2) anticipated occurrences;

   3) undesired conditions requiring system action;

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

c) the supplier and/or developer (or the multiple parties responsible for compliance) shall make available the documented results of the validation of software aspects of system safety and all pertinent documentation to the system developer to enable his product to meet the requirements of IEC 61508-1 and IEC 61508-2.

7.7.2.8 Software tools shall meet the requirements of 7.4.4.

7.7.2.9 The results of the validation of safety-related software aspects of system safety shall meet the following requirements:

a) the tests shall show that all of the specified requirements for safety-related software (see 7.2) are correctly met and the software does not perform unintended functions;

b) test cases and their results shall be documented for subsequent analysis and independent assessment (see Clause 8 of IEC 61508-1) as required by the safety integrity level;

c) the documented results of validating the software aspects of system safety shall state either (1) that the software has passed the validation or (2) the reasons for not passing the validation.

7.8 Software modification

NOTE This phase is Box 10.5 of Figure 4.

7.8.1 Objective

The objective of the requirements of this subclause is to guide corrections, enhancements or adaptations to the validated software, ensuring that the required software systematic capability is sustained.

7.8.2 Requirements

NOTE For the selection of appropriate techniques and measures (see Annexes A and B) to implement the requirements of this clause, the following properties (see Annex C for guidance on interpretation of properties, and Annex F of IEC 61508-7 for informal definitions) of the software modification should be considered:

– completeness of modification with respect to its requirements;

– correctness of modification with respect to its requirements;

– freedom from introduction of intrinsic design faults;

– avoidance of unwanted behaviour;

– verifiable and testable design;

– regression testing and verification coverage.

7.8.2.1 Prior to carrying out any software modification, software modification procedures shall be made available (see 7.16 of IEC 61508-1).

NOTE 1 Subclauses 7.8.2.1 to 7.8.2.9 apply primarily to changes occurring during the operational phase of the software. They may also apply during the programmable electronics integration and overall installation and commissioning phases (see 7.13 of IEC 61508-1).

NOTE 2 An example of a modification procedure model is shown in Figure 9 of IEC 61508-1.

7.8.2.2 A modification shall be initiated only on the issue of an authorized software modification request under the procedures specified during safety planning (see Clause 6) which details the following:

a) the hazards which may be affected;

b) the proposed modification;

c) the reasons for modification.

NOTE A request for modification could arise from, for example

- functional safety is found to be less than required by the safety requirements specification;
- systematic fault experience;

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

- new or amended safety legislation;
- modifications to the EUC or its use;
- modification to the overall safety requirements;
- analysis of operations and maintenance performance, indicating that the performance is below target;
- routine functional safety audits.

7.8.2.3 An analysis shall be carried out on the impact of the proposed software modification on the functional safety of the E/E/PE safety-related system:

a) to determine whether or not a hazard and risk analysis is required;

b) to determine which software safety lifecycle phases will need to be repeated.

7.8.2.4 The impact analysis results obtained in 7.8.2.3 shall be documented.

7.8.2.5 All modifications which have an impact on the functional safety of the E/E/PE safety-related system shall initiate a return to an appropriate phase of the software safety lifecycle. All subsequent phases shall then be carried out in accordance with the procedures specified for the specific phases in accordance with the requirements in this standard. Safety planning (see Clause 6) shall detail all subsequent activities.

NOTE It may be necessary to implement a full hazard and risk analysis, which may generate a need for different safety integrity levels than currently specified for the safety functions implemented by the E/E/PE safety-related systems.

7.8.2.6 The safety planning for the modification of safety-related software shall meet the requirements given in Clause 6 of IEC 61508-1. In particular:

a) identification of staff and specification of their required competency;

b) detailed specification for the modification;

c) verification planning;

d) scope of revalidation and testing of the modification to the extent required by the safety integrity level.

NOTE Depending on the nature of the application, involvement of domain experts may be important.

7.8.2.7 Modification shall be carried out as planned.

7.8.2.8 Details of all modifications shall be documented, including references to:

a) the modification/retrofit request;

b) the results of the impact analysis which assesses the impact of the proposed software modification on the functional safety, and the decisions taken with associated justifications;

c) software configuration management history;

d) deviation from normal operations and conditions;

e) all documented information affected by the modification activity.

7.8.2.9 Information on the details of all modifications shall be documented. The documentation shall include the re-verification and re-validation of data and results.

7.8.2.10 The assessment of the required modification or retrofit activity shall be dependent on the results of the impact analysis and the software systematic capability.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.9 Software verification

7.9.1 Objective

The objective of the requirements of this subclause is, to the extent required by the safety integrity level, to test and evaluate the outputs from a given software safety lifecycle phase to ensure correctness and consistency with respect to the inputs to that phase.

NOTE 1 This subclause considers the generic aspects of verification which are common to several safety lifecycle phases. This subclause does not place additional requirements for the testing element of verification in 7.4.7 (software module testing), 7.4.8 (software integration) and 7.5 (programmable electronics integration) because these are verification activities in themselves. Nor does this subclause require verification in addition to software validation (see 7.7), because in this standard software validation is the demonstration of conformance to the safety requirements specification. Checking whether the safety requirements specification is itself correct is carried out by domain experts.

NOTE 2 Depending on the software architecture, responsibility for the verification activity may be split between all organisations involved in the development and modification of the software.

7.9.2 Requirements

NOTE For the selection of appropriate techniques and measures (see Annexes A and B) to implement the requirements of this clause, the following properties (see Annex C for guidance on interpretation of properties, and Annex F of IEC 61508-7 for informal definitions) of the data verification should be considered:

– completeness of verification with respect to the previous phase;

– correctness of verification with respect to the previous phase (successful completion);

– repeatability;

– precisely defined verification configuration.

7.9.2.1 The verification of software shall be planned (see 7.3) concurrently with the development, for each phase of the software safety lifecycle, and shall be documented.

7.9.2.2 The software verification planning shall refer to the criteria, techniques and tools to be used in the verification activities, and shall address:

a) the evaluation of the safety integrity requirements;

b) the selection and documentation of verification strategies, activities and techniques;

c) the selection and utilisation of verification tools (test harness, special test software, input/output simulators etc.);

d) the evaluation of verification results;

e) the corrective actions to be taken.

7.9.2.3 The software verification shall be performed as planned.

NOTE Selection of techniques, measures for verification and the degree of independence of the verification activities will depend upon a number of factors and may be specified in application sector standards. The factors could include, for example:

- size of project;
- degree of complexity;
- degree of novelty of design;
- degree of novelty of technology.

7.9.2.4 Evidence shall be documented to show that the phase being verified has, in all respects, been satisfactorily completed.

7.9.2.5 After each verification, the verification documentation shall include:

a) identification of items to be verified;

b) identification of the information against which the verification has been done;

NOTE 1 Information against which the verification has been performed includes but is not limited to input from the previous lifecycle phase, design standards, coding standards and tools used.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

c) non-conformances.

NOTE 2 Examples of non-conformances include software modules, data structures, and algorithms poorly adapted to the problem.

7.9.2.6 All essential information from phase N of the software safety lifecycle needed for the correct execution of the next phase N+1 shall be available and shall be verified. Outputs from phase N include:

a) adequacy of the specification, design, or code in phase N for:
   1) functionality;
   2) safety integrity, performance and other requirements of safety planning (see Clause 6);
   3) readability by the development team;
   4) testability for further verification;
   5) safe modification to permit further evolution;
b) adequacy of the validation planning and/or tests specified for phase N for specifying and describing the design of phase N;
c) check for incompatibilities between:
   1) the tests specified in phase N, and the tests specified in the previous phase N–1;
   2) the outputs within phase N.

7.9.2.7 Subject to the choice of software development lifecycle (see 7.1), the following verification activities shall be performed:

a) verification of software safety requirements;
b) verification of software architecture;
c) verification of software system design;
d) verification of software module design;
e) verification of code;
f) verification of data;
g) verification of timing performance;
h) software module testing (see 7.4.7);
i) software integration testing (see 7.4.8);
j) programmable electronics integration testing (see 7.5);
k) software aspects of system safety validation (see 7.7).

NOTE For requirements a) to g) see below.

7.9.2.8 Verification of software safety requirements: after the software safety requirements specification has been completed, and before the next phase of software design and development begins, verification shall:

a) consider whether the software safety requirements specification adequately fulfils the E/E/PE system safety requirements specification (see 7.10 of IEC 61508-1 and 7.2 of IEC 61508-2) for functionality, safety integrity, performance, and any other requirements of safety planning;
b) consider whether the validation plan for software aspects of system safety adequately fulfils the software safety requirements specification;
c) check for incompatibilities between:
   1) the software safety requirements specification, and the E/E/PE system safety requirements specification (see 7.10 of IEC 61508-1 and 7.2 of IEC 61508-2);
   2) the software safety requirements specification, and the validation plan for software aspects of system safety.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.9.2.9 Verification of software architecture: after the software architecture design has been completed, verification shall:

a) consider whether the software architecture design adequately fulfils the software safety requirements specification;

b) consider whether the integration tests specified in the software architecture design are adequate;

c) consider whether the attributes of each major element/subsystem are adequate with reference to:

   1) feasibility of the safety performance required;

   2) testability for further verification;

   3) readability by the development and verification team;

   4) safe modification to permit further evolution.

d) check for incompatibilities between the following:

   1) the software architecture design, and the software safety requirements specification;

   2) the software architecture design and its integration tests;

   3) the software architecture design integration tests and the validation plan for software aspects of system safety.

7.9.2.10 Verification of software system design: after the software system design has been completed, verification shall:

a) consider whether the software system design (see 7.4.5) adequately fulfils the software architecture design;

b) consider whether the specified tests of the software system integration (see 7.4.5) adequately fulfil the software system design (see 7.4.5);

c) consider whether the attributes of each major element of the software system design specification (see 7.4.5) are adequate with reference to:

   1) feasibility of the safety performance required;

   2) testability for further verification;

   3) readability by the development and verification team;

   4) safe modification to permit further evolution.

   NOTE The software system integration tests may be specified as part of the software architecture integration tests.

d) check for incompatibilities between:

   1) the software system design specification (see 7.4.5), and the software architecture design;

   2) the software system design specification (see 7.4.5), and the software system integration test specification (see.4.5);

   3) the tests required by the software system integration test specification (see 7.4.5) and the software architecture integration test specification (see 7.4.3).

7.9.2.11 Verification of software module design: after the design of each software module has been completed, verification shall:

a) consider whether the software module design specification (see 7.4.5) adequately fulfils the software system design specification (see 7.4.5);

b) consider whether the software module test specification (see 7.4.5) is adequate for the software module design specification (see 7.4.5);

c) consider whether the attributes of each software module are adequate with reference to:

   1) feasibility of the safety performance required (see software safety requirements specification);

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

2) testability for further verification;

3) readability by the development and verification team;

4) safe modification to permit further evolution.

d) check for incompatibilities between:

1) the software module design specification (see 7.4.5), and the software system design specification (see 7.4.5);

2) (for each software module) the software module design specification (see 7.4.5), and the software module test specification (see 7.4.5);

3) the software module test specification (see 7.4.5), and the software system integration test specification (see 7.4.5).

7.9.2.12 Verification of code: the source code shall be verified by static methods to ensure conformance to the software module design specification (see 7.4.5), the required coding standards (see 7.4.4), and the validation plan for software aspects of system safety.

NOTE In the early phases of the software safety lifecycle, verification is static (for example inspection, review, formal proof, etc). Code verification includes such techniques as software inspections and walk-throughs. It is the combination of the results of code verification and software module testing that provides assurance that each software module satisfies its associated specification. From then onwards testing becomes the primary means of verification.

7.9.2.13 Verification of data.

a) The data structures shall be verified.

b) The application data shall be verified for:

1) consistency with the data structures;

2) completeness against the application requirements;

3) compatibility with the underlying system software (for example, sequence of execution, run-time, etc.); and

4) correctness of the data values.

c) All operational parameters shall be verified against the application requirements.

d) All plant interfaces and associated software (i.e. sensors and actuators and off-line interfaces: see 7.2.2.12) shall be verified for:

1) detection of anticipated interface failures;

2) tolerance to anticipated interface failures.

e) All communication interfaces and associated software shall be verified for an adequate level of:

1) failure detection;

2) protection against corruption;

3) data validation.

7.9.2.14 Verification of timing performance: predictability of behaviour in the time domain shall be verified.

NOTE Timing behaviour may include: performance, resources, response time, worst case execution time, thrashing, dead-lock free, run-time system.

8 Functional safety assessment

NOTE For the selection of appropriate techniques and measures (see Annexes A and B) to implement the requirements of this clause, the following properties (see Annex C for guidance on interpretation of properties, and Annex F of IEC 61508-7 for informal definitions) of the functional safety assessment should be considered:

– completeness of functional safety assessment with respect to this standard;

– correctness of functional safety assessment with respect to the design specifications (successful completion);

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

– traceable closure of all identified issues;

– the ability to modify the functional safety assessment after change without the need for extensive re-work of the assessment;

– repeatability;

– timeliness;

– precisely defined configuration.

8.1 The objective and requirements of Clause 8 of IEC 61508-1 apply to the assessment of safety-related software.

8.2 Unless otherwise stated in application sector international standards, the minimum level of independence of those carrying out the functional safety assessment shall be as specified in Clause 8 of IEC 61508-1.

8.3 An assessment of functional safety may make use of the results of the activities of Table A.10.

NOTE Selecting techniques from Annexes A and B does not guarantee by itself that the required safety integrity will be achieved (see 7.1.2.7). The assessor should also consider:

- the consistency and the complementary nature of the chosen methods, languages and tools for the whole development cycle;
- whether the developers use methods, languages and tools they fully understand;
- whether the methods, languages and tools are well-adapted to the specific problems encountered during development.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Annex A
(normative)

Guide to the selection of techniques and measures

Some of the subclauses of this standard have an associated table, for example 7.2 (software safety requirements specification) is associated with Table A.1. More detailed tables in Annex B expand upon some of the entries in the tables of Annex A. For example, Table B.2 expands on the topic of dynamic analysis and testing in Table A.5.

See IEC 61508-7 for an overview of the specific techniques and measures referenced in Annexes A and B.

With each technique or measure in the tables there is a recommendation for safety integrity levels 1 to 4. These recommendations are as follows.

HR	the technique or measure is highly recommended for this safety integrity level. If this technique or measure is not used then the rationale behind not using it should be detailed with reference to Annex C during the safety planning and agreed with the assessor.
R	the technique or measure is recommended for this safety integrity level as a lower recommendation to a HR recommendation.
---	the technique or measure has no recommendation for or against being used.
NR	the technique or measure is positively not recommended for this safety integrity level. If this technique or measure is used then the rationale behind using it should be detailed with reference to Annex C during the safety planning and agreed with the assessor.

Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. Only one of the alternate or equivalent techniques/measures has to be satisfied.

Other measures and techniques may be applied providing that the requirements and objectives have been met. See Annex C for guidance on selecting techniques.

The ranking of the techniques and measures is linked to the concept of *effectiveness* used in IEC 61508-2. For all other factors being equal, techniques which are ranked HR will be more effective in either preventing the introduction of systematic faults during software development, or (for the case of the software architecture) more effective in controlling residual faults in the software revealed during execution than techniques ranked as R.

Given the large number of factors that affect software systematic capability it is not possible to give an algorithm for combining the techniques and measures that will be correct for any given application. Guidance on a rationale for selecting specific techniques to achieve software systematic capability is given in Annex C.

For a particular application, the appropriate combination of techniques or measures are to be stated during safety planning, with appropriate techniques or measures being selected unless the note attached to the table makes other requirements.

Initial guidance in the form of two worked examples on the interpretation of the tables is given in IEC 61508-6.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table A.1 – Software safety requirements specification

(See 7.2)

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1a	Semi-formal methods	Table B.7	R	R	HR	HR
1b	Formal methods	B.2.2, C.2.4	---	R	R	HR
2	Forward traceability between the system safety requirements and the software safety requirements	C.2.11	R	R	HR	HR
3	Backward traceability between the safety requirements and the perceived safety needs	C.2.11	R	R	HR	HR
4	Computer-aided specification tools to support appropriate techniques/measures above	B.2.4	R	R	HR	HR

NOTE 1 The software safety requirements specification will always require a description of the problem in natural language and any necessary mathematical notation that reflects the application.

NOTE 2 The table reflects additional requirements for specifying the software safety requirements clearly and precisely.

NOTE 3 See Table C.1.

NOTE 4 The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

--`,``,,`````````,,``,`,,``,`,`-`-`,,`,`,`,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table A.2 – Software design and development – software architecture design

(see 7.4.3)

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4
	Architecture and design feature					
1	Fault detection	C.3.1	---	R	HR	HR
2	Error detecting codes	C.3.2	R	R	R	HR
3a	Failure assertion programming	C.3.3	R	R	R	HR
3b	Diverse monitor techniques (with independence between the monitor and the monitored function in the same computer)	C.3.4	---	R	R	----
3c	Diverse monitor techniques (with separation between the monitor computer and the monitored computer)	C.3.4	---	R	R	HR
3d	Diverse redundancy, implementing the same software safety requirements specification	C.3.5	---	---	---	R
3e	Functionally diverse redundancy, implementing different software safety requirements specification	C.3.5	---	---	R	HR
3f	Backward recovery	C.3.6	R	R	---	NR
3g	Stateless software design (or limited state design)	C.2.12	---	---	R	HR
4a	Re-try fault recovery mechanisms	C.3.7	R	R	---	---
4b	Graceful degradation	C.3.8	R	R	HR	HR
5	Artificial intelligence - fault correction	C.3.9	---	NR	NR	NR
6	Dynamic reconfiguration	C.3.10	---	NR	NR	NR
7	Modular approach	Table B.9	HR	HR	HR	HR
8	Use of trusted/verified software elements (if available)	C.2.10	R	HR	HR	HR
9	Forward traceability between the software safety requirements specification and software architecture	C.2.11	R	R	HR	HR
10	Backward traceability between the software safety requirements specification and software architecture	C.2.11	R	R	HR	HR
11a	Structured diagrammatic methods **	C.2.1	HR	HR	HR	HR
11b	Semi-formal methods **	Table B.7	R	R	HR	HR
11c	Formal design and refinement methods **	B.2.2, C.2.4	---	R	R	HR
11d	Automatic software generation	C.4.6	R	R	R	R
12	Computer-aided specification and design tools	B.2.4	R	R	HR	HR
13a	Cyclic behaviour, with guaranteed maximum cycle time	C.3.11	R	HR	HR	HR
13b	Time-triggered architecture	C.3.11	R	HR	HR	HR
13c	Event-driven, with guaranteed maximum response time	C.3.11	R	HR	HR	-
14	Static resource allocation	C.2.6.3	-	R	HR	HR
15	Static synchronisation of access to shared resources	C.2.6.3	-	-	R	HR

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

NOTE 1 Some of the methods given in Table A.2 are about design concepts, others are about how the design is represented.

NOTE 2 The measures in this table concerning fault tolerance (control of failures) should be considered with the requirements for architecture and control of failures for the hardware of the programmable electronics in IEC 61508-2.

NOTE 3 See Table C.2.

NOTE 4 The group 13 measures apply only to systems and software with safety timing requirements.

NOTE 5 Measure 14. The use of dynamic objects (for example on the execution stack or on a heap) may impose requirements on both available memory and also execution time. Measure 14 does not need to be applied if a compiler is used which ensures a) that sufficient memory for all dynamic variables and objects will be allocated before runtime, or which guarantees that in case of memory allocation error, a safe state is achieved; b) that response times meet the requirements.

NOTE 6 Measure 4a. Re-try fault recovery is often appropriate at any SIL but a limit should be set on the number of retries.

NOTE 7 The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

** Group 11, “Structured methods”. Use measure 11a only if 11b is not suited to the domain for SIL 3+4.

Table A.3 – Software design and development – support tools and programming language

(See 7.4.4)

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	Suitable programming language	C.4.5	HR	HR	HR	HR
2	Strongly typed programming language	C.4.1	HR	HR	HR	HR
3	Language subset	C.4.2	---	---	HR	HR
4a	Certified tools and certified translators	C.4.3	R	HR	HR	HR
4b	Tools and translators: increased confidence from use	C.4.4	HR	HR	HR	HR

NOTE 1 See Table C.3.

NOTE 2 The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table A.4 – Software design and development – detailed design

(See 7.4.5 and 7.4.6)

(Includes software system design, software module design and coding)

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1a	Structured methods **	C.2.1	HR	HR	HR	HR
1b	Semi-formal methods **	Table B.7	R	HR	HR	HR
1c	Formal design and refinement methods **	B.2.2, C.2.4	---	R	R	HR
2	Computer-aided design tools	B.3.5	R	R	HR	HR
3	Defensive programming	C.2.5	---	R	HR	HR
4	Modular approach	Table B.9	HR	HR	HR	HR
5	Design and coding standards	C.2.6 Table B.1	R	HR	HR	HR
6	Structured programming	C.2.7	HR	HR	HR	HR
7	Use of trusted/verified software elements (if available)	C.2.10	R	HR	HR	HR
8	Forward traceability between the software safety requirements specification and software design	C.2.11	R	R	HR	HR

NOTE 1 See Table C.4.

NOTE 2 There is still debate about the suitability of OO software development for safety-related systems. See Annex G of IEC 61508-7 for guidance on object oriented architecture and design.

NOTE 3 The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

** Group 1, “Structured methods”. Use measure 1a only if 1b is not suited to the domain for SIL 3+4.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table A.5 – Software design and development – software module testing and integration

(See 7.4.7 and 7.4.8)

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	Probabilistic testing	C.5.1	---	R	R	R
2	Dynamic analysis and testing	B.6.5 Table B.2	R	HR	HR	HR
3	Data recording and analysis	C.5.2	HR	HR	HR	HR
4	Functional and black box testing	B.5.1 B.5.2 Table B.3	HR	HR	HR	HR
5	Performance testing	Table B.6	R	R	HR	HR
6	Model based testing	C.5.27	R	R	HR	HR
7	Interface testing	C.5.3	R	R	HR	HR
8	Test management and automation tools	C.4.7	R	HR	HR	HR
9	Forward traceability between the software design specification and the module and integration test specifications	C.2.11	R	R	HR	HR
10	Formal verification	C.5.12	---	---	R	R

NOTE 1 Software module and integration testing are verification activities (see Table B.9).

NOTE 2 See Table C.5.

NOTE 3 Technique 9. Formal verification may reduce the amount and extent of module and integration testing required.

NOTE 4 The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

Table A.6 – Programmable electronics integration (hardware and software)

(See 7.5)

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	Functional and black box testing	B.5.1 B.5.2 Table B.3	HR	HR	HR	HR
2	Performance testing	Table B.6	R	R	HR	HR
3	Forward traceability between the system and software design requirements for hardware/software integration and the hardware/software integration test specifications	C.2.11	R	R	HR	HR

NOTE 1 Programmable electronics integration is a verification activity (see Table A.9).

NOTE 2 See Table C.6.

NOTE 3 The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table A.7 – Software aspects of system safety validation

(See 7.7)

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	Probabilistic testing	C.5.1	---	R	R	HR
2	Process simulation	C.5.18	R	R	HR	HR
3	Modelling	Table B.5	R	R	HR	HR
4	Functional and black-box testing	B.5.1 B.5.2 Table B.3	HR	HR	HR	HR
5	Forward traceability between the software safety requirements specification and the software safety validation plan	C.2.11	R	R	HR	HR
6	Backward traceability between the software safety validation plan and the software safety requirements specification	C.2.11	R	R	HR	HR

NOTE 1 See Table C.7.

NOTE 2 The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

Table A.8 – Modification

(See 7.8)

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	Impact analysis	C.5.23	HR	HR	HR	HR
2	Reverify changed software module	C.5.23	HR	HR	HR	HR
3	Reverify affected software modules	C.5.23	R	HR	HR	HR
4a	Revalidate complete system	Table A.7	---	R	HR	HR
4b	Regression validation	C.5.25	R	HR	HR	HR
5	Software configuration management	C.5.24	HR	HR	HR	HR
6	Data recording and analysis	C.5.2	HR	HR	HR	HR
7	Forward traceability between the Software safety requirements specification and the software modification plan (including reverification and revalidation)	C.2.11	R	R	HR	HR
8	Backward traceability between the software modification plan (including reverification and revalidation)and the software safety requirements specification	C.2.11	R	R	HR	HR

NOTE 1 See Table C.8.

NOTE 2 Techniques group 4. Impact analysis is a necessary part of regression validation. See IEC 61508-7.

NOTE 3 The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table A.9 – Software verification

(See 7.9)

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	Formal proof	C.5.12	---	R	R	HR
2	Animation of specification and design	C.5.26	R	R	R	R
3	Static analysis	B.6.4 Table B.8	R	HR	HR	HR
4	Dynamic analysis and testing	B.6.5 Table B.2	R	HR	HR	HR
5	Forward traceability between the software design specification and the software verification (including data verification) plan	C.2.11	R	R	HR	HR
6	Backward traceability between the software verification (including data verification) plan and the software design specification	C.2.11	R	R	HR	HR
7	Offline numerical analysis	C.2.13	R	R	HR	HR
Software module testing and integration		See Table A.5				
Programmable electronics integration testing		See Table A.6				
Software system testing (validation)		See Table A.7				

NOTE 1 For convenience all verification activities have been drawn together under this table. However, this does not place additional requirements for the dynamic testing element of verification in Table A.5 and Table A.6 which are verification activities in themselves. Nor does this table require verification testing in addition to software validation (see Table B.7), which in this standard is the demonstration of conformance to the safety requirements specification (end-end verification).

NOTE 2 Verification crosses the boundaries of IEC 61508-1, IEC 61508-2 and IEC 61508-3. Therefore the first verification of the safety-related system is against the earlier system level specifications.

NOTE 3 In the early phases of the software safety lifecycle verification is static, for example inspection, review, formal proof. When code is produced dynamic testing becomes possible. It is the combination of both types of information that is required for verification. For example code verification of a software module by static means includes such techniques as software inspections, walk-throughs, static analysis, formal proof. Code verification by dynamic means includes functional testing, white-box testing, statistical testing. It is the combination of both types of evidence that provides assurance that each software module satisfies its associated specification.

NOTE 4 See Table C.9.

NOTE 5 The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table A.10 – Functional safety assessment

(see Clause 8)

Assessment/Technique *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	Checklists	B.2.5	R	R	R	R
2	Decision/truth tables	C.6.1	R	R	R	R
3	Failure analysis	Table B.4	R	R	HR	HR
4	Common cause failure analysis of diverse software (if diverse software is actually used)	C.6.3	---	R	HR	HR
5	Reliability block diagram	C.6.4	R	R	R	R
6	Forward traceability between the requirements of Clause 8 and the plan for software functional safety assessment	C.2.11	R	R	HR	HR
NOTE 1 See Table C.10. NOTE 2 The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.						
* Appropriate techniques/measures shall be selected according to the safety integrity level.						

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Annex B
(informative)

Detailed tables

Table B.1 – Design and coding standards

(Referenced by Table A.4)

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	Use of coding standard to reduce likelihood of errors	C.2.6.2	HR	HR	HR	HR
2	No dynamic objects	C.2.6.3	R	HR	HR	HR
3a	No dynamic variables	C.2.6.3	---	R	HR	HR
3b	Online checking of the installation of dynamic variables	C.2.6.4	---	R	HR	HR
4	Limited use of interrupts	C.2.6.5	R	R	HR	HR
5	Limited use of pointers	C.2.6.6	---	R	HR	HR
6	Limited use of recursion	C.2.6.7	---	R	HR	HR
7	No unstructured control flow in programs in higher level languages	C.2.6.2	R	HR	HR	HR
8	No automatic type conversion	C.2.6.2	R	HR	HR	HR

NOTE 1 Measures 2, 3a and 5. The use of dynamic objects (for example on the execution stack or on a heap) may impose requirements on both available memory and also execution time. Measures 2, 3a and 5 do not need to be applied if a compiler is used which ensures a) that sufficient memory for all dynamic variables and objects will be allocated before runtime, or which guarantees that in case of memory allocation error, a safe state is achieved; b) that response times meet the requirements.

NOTE 2 See Table C.11.

NOTE 3 The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table B.2 – Dynamic analysis and testing

(Referenced by Tables A.5 and A.9)

Technique/Measure *		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	Test case execution from boundary value analysis	C.5.4	R	HR	HR	HR
2	Test case execution from error guessing	C.5.5	R	R	R	R
3	Test case execution from error seeding	C.5.6	---	R	R	R
4	Test case execution from model-based test case generation	C.5.27	R	R	HR	HR
5	Performance modelling	C.5.20	R	R	R	HR
6	Equivalence classes and input partition testing	C.5.7	R	R	R	HR
7a	Structural test coverage (entry points) 100 % **	C.5.8	HR	HR	HR	HR
7b	Structural test coverage (statements) 100 %**	C.5.8	R	HR	HR	HR
7c	Structural test coverage (branches) 100 %**	C.5.8	R	R	HR	HR
7d	Structural test coverage (conditions, MC/DC) 100 %**	C.5.8	R	R	R	HR

NOTE 1 The analysis for the test cases is at the subsystem level and is based on the specification and/or the specification and the code.

NOTE 2 See Table C.12.

NOTE 3 The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

** Where 100 % coverage cannot be achieved (e.g. statement coverage of defensive code), an appropriate explanation should be given.

Table B.3 – Functional and black-box testing

(Referenced by Tables A.5, A.6 and A.7)

Technique/Measure *		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	Test case execution from cause consequence diagrams	B.6.6.2	---	---	R	R
2	Test case execution from model-based test case generation	C.5.27	R	R	HR	HR
3	Prototyping/animation	C.5.17	---	---	R	R
4	Equivalence classes and input partition testing, including boundary value analysis	C.5.7 C.5.4	R	HR	HR	HR
5	Process simulation	C.5.18	R	R	R	R

NOTE 1 The analysis for the test cases is at the software system level and is based on the specification only.

NOTE 2 The completeness of the simulation will depend upon the safety integrity level, complexity and application.

NOTE 3 See Table C.13.

NOTE 4 The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table B.4 – Failure analysis

(Referenced by Table A.10)

Technique/Measure *		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1a	Cause consequence diagrams	B.6.6.2	R	R	R	R
1b	Event tree analysis	B.6.6.3	R	R	R	R
2	Fault tree analysis	B.6.6.5	R	R	R	R
3	Software functional failure analysis	B.6.6.4	R	R	R	R

NOTE 1 Preliminary hazard analysis should have already taken place in order to categorize the software into the most appropriate safety integrity level.

NOTE 2 See Table C.14.

NOTE 3 The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

Table B.5 – Modelling

(referenced by Table A.7)

Technique/Measure *		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	Data flow diagrams	C.2.2	R	R	R	R
2a	Finite state machines	B.2.3.2	---	R	HR	HR
2b	Formal methods	B.2.2, C.2.4	---	R	R	HR
2c	Time Petri nets	B.2.3.3	---	R	HR	HR
3	Performance modelling	C.5.20	R	HR	HR	HR
4	Prototyping/animation	C.5.17	R	R	R	R
5	Structure diagrams	C.2.3	R	R	R	HR

NOTE 1 If a specific technique is not listed in the table, it should not be assumed that it is excluded from consideration. It should conform to this standard.

NOTE 2 Quantification of probabilities is not required.

NOTE 3 See Table C.15.

NOTE 4 The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table B.6 – Performance testing

(referenced by Tables A.5 and A.6)

Technique/Measure *		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	Avalanche/stress testing	C.5.21	R	R	HR	HR
2	Response timings and memory constraints	C.5.22	HR	HR	HR	HR
3	Performance requirements	C.5.19	HR	HR	HR	HR
NOTE 1 See Table C.16. NOTE 2 The references (which are informative, not normative) "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.						
* Appropriate techniques/measures shall be selected according to the safety integrity level.						

Table B.7 – Semi-formal methods

(Referenced by Tables A.1, A.2 and A.4)

Technique/Measure *		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	Logic/function block diagrams	See Note 1	R	R	HR	HR
2	Sequence diagrams	see Note 1	R	R	HR	HR
3	Data flow diagrams	C.2.2	R	R	R	R
4a	Finite state machines/state transition diagrams	B.2.3.2	R	R	HR	HR
4b	Time Petri nets	B.2.3.3	R	R	HR	HR
5	Entity-relationship-attribute data models	B.2.4.4	R	R	R	R
6	Message sequence charts	C.2.14	R	R	R	R
7	Decision/truth tables	C.6.1	R	R	HR	HR
8	UML	C.3.12	R	R	R	R
NOTE 1 Logic/function block diagrams and sequence diagrams are described in IEC 61131-3. NOTE 2 See Table C.17. NOTE 3 The references "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.						
* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.						

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table B.8 – Static analysis

(Referenced by Table A.9)

Technique/Measure *		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	Boundary value analysis	C.5.4	R	R	HR	HR
2	Checklists	B.2.5	R	R	R	R
3	Control flow analysis	C.5.9	R	HR	HR	HR
4	Data flow analysis	C.5.10	R	HR	HR	HR
5	Error guessing	C.5.5	R	R	R	R
6a	Formal inspections, including specific criteria	C.5.14	R	R	HR	HR
6b	Walk-through (software)	C.5.15	R	R	R	R
7	Symbolic execution	C.5.11	---	---	R	R
8	Design review	C.5.16	HR	HR	HR	HR
9	Static analysis of run time error behaviour	B.2.2, C.2.4	R	R	R	HR
10	Worst-case execution time analysis	C.5.20	R	R	R	R

NOTE 1 See Table C.18.

NOTE 2 The references "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

Table B.9 – Modular approach

(Referenced by Table A.4)

Technique/Measure *		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	Software module size limit	C.2.9	HR	HR	HR	HR
2	Software complexity control	C.5.13	R	R	HR	HR
3	Information hiding/encapsulation	C.2.8	R	HR	HR	HR
4	Parameter number limit / fixed number of subprogram parameters	C.2.9	R	R	R	R
5	One entry/one exit point in subroutines and functions	C.2.9	HR	HR	HR	HR
6	Fully defined interface	C.2.9	HR	HR	HR	HR

NOTE 1 See Table C.19.

NOTE 2 The references "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. No single technique is likely to be sufficient. All appropriate techniques shall be considered.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Annex C
(informative)

Properties for software systematic capability

C.1 Introduction

Given the large number of factors that affect software systematic capability it is not possible to give an algorithm for combining the techniques and measures that will be correct for any given application. The purpose of Annex C is:

– to give guidance on selecting specific techniques from Annexes A and B to achieve software systematic capability;

– to outline a rationale for justifying the use of techniques that are not explicitly listed in Annexes A and B.

Annex C is supplementary to Annexes A and B tables.

C.1.1 Structure of Annex C, relating to Annexes A and B

The outputs from each phase of the software safety lifecycle are defined in Table 1. For example, consider the software safety requirements specification.

Table A.1 (“Software safety requirements specification”) of Annex A recommends specific techniques for developing the software safety requirements specification.

Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1a	Semi-formal methods	Table B.7	R	R	HR	HR
1b	Formal methods	B.2.2, C.2.4	---	R	R	HR
2	Forward traceability between the system safety requirements and the software safety requirements	C.2.11	R	R	HR	HR
3	Backward traceability between the safety requirements and the perceived safety needs	C.2.11	R	R	HR	HR
4	Computer-aided specification tools to support appropriate techniques/measures above	B.2.4	R	R	HR	HR

Annex C Table C.1 (“Properties for systematic safety integrity – Software safety requirements specification”) states that the software safety requirements specification is characterized by the following desirable properties (which are informally defined in Annex F of IEC 61508-7):

Properties					
Completeness with respect to the safety needs to be addressed by software	Correctness with respect to the safety needs to be addressed by software	Freedom from intrinsic specification faults, including freedom from ambiguity	Understandability of safety requirements	Freedom from adverse interference of non-safety functions with the safety needs to be addressed by software	Capability of providing a basis for verification and validation

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Annex C Table C.1 also ranks on an informal scale R1/R2/R3 the effectiveness of specific techniques in achieving these desirable properties.

Technique/ Measure		Properties					
		Completeness with respect to the safety needs to be addressed by software	Correctness with respect to the safety needs to be addressed by software	Freedom from intrinsic specification faults, including freedom from ambiguity	Understandability of safety requirements	Freedom from adverse interference of non-safety functions with the safety needs to be addressed by software	Capability of providing a basis for verification and validation
1a	Semi-formal methods	R1 Application-friendly or domain specific specification method and notation used by domain experts	R1 Application-friendly or domain specific specification method and notation used by domain experts R2 Verification of specification according to coverage criteria	R1 Method and notation that helps avoid or detect internal inconsistency, missing behaviour or mathematically inconsistent expressions. R2 Verification of specification according to coverage criteria R3 Verification of specification based on systematic analysis, and / or systematic avoidance of particular types of intrinsic specification faults	R1 Defined notation that restricts opportunity for misunderstanding R2 Application of complexity limits in specification	_	R2 Defined notation that reduces ambiguity in specification

The confidence that can be placed in the software safety requirements specification as a basis for safe software depends on the rigour of the techniques by which the desirable properties of the software safety requirements specification have been achieved. The rigour of a technique is informally ranked on a scale R1 to R3, where R1 is the least rigorous and R3 the most rigorous.

R1	without objective acceptance criteria, or with limited objective acceptance criteria. E.g., black-box testing based on judgement, field trials.
R2	with objective acceptance criteria that can give a high level of confidence that the required property is achieved (exceptions to be identified & justified); e.g., test or analysis techniques with coverage metrics, coverage of checklists.
R3	with objective, systematic reasoning that the required property is achieved. E.g. formal proof, demonstrated adherence to architectural constraints that guarantee the property.
_	this technique is not relevant to this property.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

A technique may achieve one of several R1/R2/R3 rankings relating to a particular property, depending on the level of rigour that the technique satisfies.

Technique/ Measure		Properties					
		Completeness with respect to the safety needs to be addressed by software	Correctness with respect to the safety needs to be addressed by software	Freedom from intrinsic specification faults, including freedom from ambiguity	Understandability of safety requirements	Freedom from adverse interference of non-safety functions with the safety needs to be addressed by software	Capability of providing a basis for verification and validation
1a	Semi-formal methods				R1 Defined notation that restricts opportunity for misunderstanding R2 Application of complexity limits in specification		

In this example, a semi-formal method achieves rigour R1 by providing a restricted notation that improves accurate expression, and achieves R2 by further restricting the complexity of specification which might otherwise cause confusion.

C.1.2 Method of use – 1

For guidance purposes, if it can be convincingly demonstrated that the desirable properties have been achieved in the development of the software safety requirements specification, then confidence is justified that the software safety requirements specification is an adequate basis for developing software that has sufficient systematic safety integrity.

Annex C Table C.1 says that each of the Annex A Table A.1 techniques typically achieves, to a greater or lesser extent, one or more of the above Table C.1 properties that are relevant to the software safety requirements specification.

However, it is important to note that although Annex A Table A.1 recommends specific techniques, these recommendations are not prescriptive, and in fact Annex A states clearly that “Given the large number of factors that affect software systematic capability it is not possible to give an algorithm for combining the techniques and measures that will be correct for any given application”.

In practice the techniques by which the software safety requirements specification is developed are selected subject to several practical constraints (see 7.1.2.7) in addition to the inherent capabilities of the techniques. Such constraints may include:

– the consistency and the complementary nature of the chosen methods, languages and tools for the whole development cycle;

– whether the developers use methods, languages and tools they fully understand;

– whether the methods, languages and tools are well-adapted to the specific problems encountered during development.

Table C.1 may be used to compare the relative effectiveness of the specific Annex A Table A.1 techniques in achieving the desirable properties of the software safety requirements specification lifecycle, while at the same time factoring in the practical constraints of the particular development project.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

For example, a formal method is capable of giving a better basis (R3) for verification and validation than is a semi-formal method (R2), but other project constraints (e.g. the availability of sophisticated computer support tools, or the very specialized expressiveness of a formal notation) may favour a semi-formal approach.

In this way, the Table C.1 desirable properties can provide the basis of a reasoned and practical comparison of the alternative techniques that Annex A Table A.1 recommends for developing the software safety requirements specification. Or more generally, a reasoned selection from the several alternative techniques recommended by Annex A for a particular lifecycle phase can be made by considering the desirable properties listed in the corresponding Annex C table.

But note carefully that due to the nature of systematic behaviour, these Annex C properties may not be achievable or demonstrable with the highest rigour. Rather, they are goals to be aimed for. Their achievement may even necessitate trade-offs between different properties e.g. between defensive design and simplicity.

Finally, in addition to defining R1/R2/R3 criteria, it is useful for guidance purposes to make an informal link between (1) the increasing level of rigour of the R1 to R3 progression and (2) an increased confidence in the correctness of the software. As a general and informal recommendation, the following minimum levels of rigour should be aimed for when Annex A requires the corresponding SIL performance:

SIL	Rigour R
1 / 2	R1
3	R2 where available
4	highest rigour available

C.1.3 Method of use – 2

Although Annex A recommends specific techniques, it is also permitted to apply other measures and techniques, providing that the requirements and objectives of the lifecycle phase have been met.

It has already been noted that many factors affect software systematic capability, and it is not possible to give an algorithm for selecting and combining the techniques in a way that is guaranteed in any given application to achieve the desirable properties.

There may be several effective ways to achieve the desirable properties, and it should be recognized that system developers may be able to provide alternative evidence. The information in these Annex C tables can be used as the basis of a reasoned argument to justify the selection of techniques other than those given in the Annex A tables.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

C.2 Properties for systematic safety integrity

The guidance here and in IEC 61508-7 indicates specific techniques for achieving the systematic safety integrity properties and for generating convincing evidence. Where a method does not contribute to the achievement of a property, this is shown in the following tables by a dash. Where a method may have adverse effects on some properties and positive effects on others, a note is provided in the relevant table below.

Table C.1 – Properties for systematic safety integrity – Software safety requirements specification

(See 7.2. Referenced by Table A.1)

Technique/Measure		Properties					
		Completeness with respect to the safety needs to be addressed by software	**Correctness with respect to the safety needs to be addressed by software**	**Freedom from intrinsic specification faults, including freedom from ambiguity**	**Understandability of safety requirements**	**Freedom from adverse interference of non-safety functions with the safety needs to be addressed by software**	**Capability of providing a basis for verification and validation**
1a	Semi-formal methods	R1 Application-friendly or domain specific specification method and notation used by domain experts	R1 Application-friendly or domain specific specification method and notation used by domain experts R2 Verification of specification according to coverage criteria	R1 Method and notation that helps avoid or detect internal inconsistency, missing behaviour or mathematically inconsistent expressions. R2 Verification of specification according to coverage criteria R3 Verification of specification based on systematic analysis, and / or systematic avoidance of particular types of intrinsic specification faults	R1 Defined notation that restricts opportunity for misunderstanding R2 Application of complexity limits in specification	–	R2 Defined notation that reduces ambiguity in specification

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Measure		Properties					
		Completeness with respect to the safety needs to be addressed by software	**Correctness with respect to the safety needs to be addressed by software**	**Freedom from intrinsic specification faults, including freedom from ambiguity**	**Understandability of safety requirements**	**Freedom from adverse interference of non-safety functions with the safety needs to be addressed by software**	**Capability of providing a basis for verification and validation**
1b	Formal methods	R1 Application-friendly or domain specific specification method and notation used by domain experts	R1 Application-friendly or domain specific specification method and notation used by domain experts. R2 Verification of specification according to coverage criteria R3 Guarantee of correctness on limited aspects of behaviour	R1 Method and notation that help avoid or detect internal inconsistency, missing behaviour or mathematically inconsistent expressions. R2 Verification of specification according to coverage criteria R3 Verification of specification based on systematic analysis, and / or Systematic avoidance of particular types of intrinsic specification faults	– Note: May complicate the achievement of this property if the method is not application-friendly or domain specific.	–	R3 Reduces ambiguity in specification.
2	Forward traceability between the system safety requirements specification and the software safety requirements	R1 Confidence that the Software safety requirements specification addresses the system safety requirements	–	–	–	–	–
3	Backward traceability between the software safety requirements specification and the perceived safety needs	–	R1 Confidence that the software safety requirements specification contains no unnecessary complexity	–	RI Traceability to the EUC safety needs enhances understandability	R1	R1

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Measure		Properties					
		Completeness with respect to the safety needs to be addressed by software	**Correctness with respect to the safety needs to be addressed by software**	**Freedom from intrinsic specification faults, including freedom from ambiguity**	**Understandability of safety requirements**	**Freedom from adverse interference of non-safety functions with the safety needs to be addressed by software**	**Capability of providing a basis for verification and validation**
4	Computer-aided specification tools to support appropriate techniques/measures above	R1 Encapsulation of domain knowledge of the EUC and of the software environment R2 If checklist of issues to be taken into consideration is defined, justified and covered	R1 Functional simulation techniques R2 Functional simulation according to defined and justified coverage criteria	R2 Semantic and syntactic checks to ensure that the relevant rules are satisfied	R1 Animation of, or browsing through the specification	R1 Identification of safety and non-safety functions	R1 Assists traceability and coverage R2 Measurement of traceability and coverage

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Table C.2 – Properties for systematic safety integrity – Software design and development – software Architecture Design

(See 7.4.3. Referenced by Table A.2)

Technique/Measure		Properties							
		Completeness with respect to software safety requirements specification	**Correctness with respect to software safety requirements specification**	**Freedom from intrinsic design faults**	**Simplicity and understandability**	**Predictability of behaviour**	**Verifiable and testable design**	**Fault tolerance**	**Defence against common cause failure from external events**
1	Fault detection	_	_	_	_ May complicate the achievement of this property	R1 Logical program flow monitoring provides for predictability	_	R1 (R2 if coverage targets are defined, justified and met)	R1 or _
2	Error detecting codes	_	_	_	_ May complicate the achievement of this property.	_ May complicate the achievement of this property.	_	R1 (R2 if coverage targets are defined, justified and met) Effective for specific application areas e.g. data comms	R1 Effective for specific application areas e.g. data comms
3a	Failure assertion programming	_	R2 Post-assertions may check compliance with detailed requirements	_	R2 Pre-assertions limit the input space	R2 Post-assertions check for expected / acceptable outputs	R2 Pre-assertions limit the input space and hence the required test space	R3 Effective for the targeted failures	R3 Effective for the targeted failures
3b	Diverse monitor techniques (with independence between the monitor and the monitored function in the same computer)	_	_	R2 Diverse monitor implements only the minimum safety requirements	R2 Diverse monitor provides for implicit diversity	R2 Diverse monitor implements in a simple manner only the minimum safety requirements	R2 Diverse monitor implements only the minimum safety requirements	R1 (R2 if coverage targets are defined, justified and met)	R1 (R2 if coverage targets are defined, justified and met)

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Measure		Properties							
		Completeness with respect to software safety requirements specification	**Correctness with respect to software safety requirements specification**	**Freedom from intrinsic design faults**	**Simplicity and understandability**	**Predictability of behaviour**	**Verifiable and testable design**	**Fault tolerance**	**Defence against common cause failure from external events**
3c	Diverse monitor techniques (with separation between the monitor computer and the monitored computer)	-	_	R2 Diverse monitor implements only the minimum safety requirements	R2 Diverse monitor provides for implicit diversity	R2 Diverse monitor implements in a simple manner only the minimum safety requirements)	R2 Diverse monitor implements only the minimum safety requirements	R1 (R2 if coverage targets are defined, justified and met)	R1 (R2 if coverage targets are defined, justified and met)
3d	Diverse redundancy, implementing the same software safety requirements specification	_	_	_	_ Note: May complicate the achievement of this property if done within the same executable software.	_	_	R1 If the failure of one program does not adversely affect the others R2 If coverage targets are defined, justified and met Does not protect against requirements specification faults	R1 If the failure of one program does not adversely affect the others R2 If coverage targets are defined, justified and met Does not protect against requirements specification faults

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Technique/Measure		Properties							
		Completeness with respect to software safety requirements specification	**Correctness with respect to software safety requirements specification**	**Freedom from intrinsic design faults**	**Simplicity and understandability**	**Predictability of behaviour**	**Verifiable and testable design**	**Fault tolerance**	**Defence against common cause failure from external events**
3e	Functionally diverse redundancy, implementing different software safety requirements specification. This will typically require sensors operating on different physical principles	–	–	R1	– Note: May complicate the achievement of this property if done within the same executable software.	–	–	R1 If the failure of one program does not adversely affect the others.	R1 If the failure of one program does not adversely affect the others. Protects against specification faults
3f	Backward recovery	–	–	– Note: May complicate the achievement of this property.	–	– Note: May complicate the achievement of this property.	–	R2	R1 (R2 if coverage targets are defined, justified and met)
3g	Stateless design (or limited state design)	R2 Provided safety requirements are also stateless or limited state	R2 Provided safety requirements are also stateless or limited state	R2 Provided safety requirements are also stateless or limited state	R1 R2 If limits are defined, justified and met regarding the possible number of states	R1 R2 If limits are defined, justified and met regarding the possible number of states	R1 R2 If targets are defined, justified and met for the verification / test coverage of the possible states	R1 If this leads to a self-healing design R2 If targets are defined, justified and met for self-healing	R1 If this leads to a self-healing design R2 If targets are defined, justified and met for self-healing
4a	Re-try fault recovery mechanisms	–	–	–	–	May complicate the achievement of this property	–	R1 (R2 if coverage targets are defined, justified and met)	R1 (R2 if coverage targets are defined, justified and met)

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Measure		Properties							
		Completeness with respect to software safety requirements specification	**Correctness with respect to software safety requirements specification**	**Freedom from intrinsic design faults**	**Simplicity and understandability**	**Predictability of behaviour**	**Verifiable and testable design**	**Fault tolerance**	**Defence against common cause failure from external events**
4b	Graceful degradation	–	–	– Note: May complicate the achievement of this property.	–	–	–	R1 R2 if coverage targets are defined, justified and met	R1 R2 if coverage targets are defined, justified and met
5	Artificial intelligence - fault correction	–	– Note: May complicate the achievement of this property.	– Note: May complicate the achievement of this property.	– Note: May complicate the achievement of this property.	– Note: May complicate the achievement of this property.	– Note: May complicate the achievement of this property.	–	–
6	Dynamic reconfiguration	–	– Note: May complicate the achievement of this property.	– Note: May complicate the achievement of this property.	– Note: May complicate the achievement of this property.	– Note: May complicate the achievement of this property.	– Note: May complicate the achievement of this property.	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Measure		Properties							
		Completeness with respect to software safety requirements specification	**Correctness with respect to software safety requirements specification**	**Freedom from intrinsic design faults**	**Simplicity and understandability**	**Predictability of behaviour**	**Verifiable and testable design**	**Fault tolerance**	**Defence against common cause failure from external events**
7	Modular approach	–	R1 R2 R2 is achieved If modularity targets are defined, justified and met. Otherwise, only R1 is achieved.	R1 If freedom from particular types of intrinsic design faults can be verified independently for each module R3 If freedom from particular types of intrinsic design faults can be supported by a rigorous reasoning based on modular design	R1 R2 If modularity targets are defined, justified and met	R1 R2 If modularity targets are defined, justified and met	R1 R2 If modularity targets are defined, justified and met	R1 If modules not affected by the failure of a module contribute in mitigation / recovery R3 If tolerance to particular faults can be supported by a rigorous reasoning	R1 If modules that can be influenced by external events that can affect multiple channels concurrently, are identified and subject to thorough verification R3 If tolerance to particular external events can be supported by a rigorous reasoning

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Technique/Measure		Properties							
		Completeness with respect to software safety requirements specification	**Correctness with respect to software safety requirements specification**	**Freedom from intrinsic design faults**	**Simplicity and understandability**	**Predictability of behaviour**	**Verifiable and testable design**	**Fault tolerance**	**Defence against common cause failure from external events**
8	Use of trusted/verified software modules and elements (if available)	–	R1 R2 R3 If the element significantly contributes to particular safety requirements, and is correctly used	R1 R2 R3 Re-uses proven elements. Such capability shall be justified for the element	R1 Modular approach decomposes overall complexity into understandable units	R1 R2 R3 Reuses proven elements	–	R1R2 R2 If fault tolerance capabilities are readily provided by the element and are correctly used, or if a fault tolerance layer is built around the element	R1R2 R2 If defences against external events that could affect concurrently multiple channels are readily provided by the element and are correctly used, or if a defensive layer is built around the element
9	Forward traceability between the software safety requirements specification and software architecture	R1 Confidence that the architecture addresses the software safety requirements	–	–	–	–	–	–	–
10	Backward traceability between the software architecture and the software safety requirements specification	–	R1 Confidence that the architecture contains no unnecessary complexity	–	–	–	–	–	–
11a	Structured diagrammatic methods	–	R1	–	R1 (Graphical descriptions are easier to understand)	–	R1 (Structured designs are easier to verify and test)	–	–

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Measure		Properties							
		Completeness with respect to software safety requirements specification	**Correctness with respect to software safety requirements specification**	**Freedom from intrinsic design faults**	**Simplicity and understandability**	**Predictability of behaviour**	**Verifiable and testable design**	**Fault tolerance**	**Defence against common cause failure from external events**
11b	Semi-formal methods	R1 An application-friendly or domain specific specification method and notation	R1 An application-friendly or domain specific specification method and notation	R2 Can detect internal inconsistency or missing behaviour or mathematically inconsistent expressions	–	R2 (Provides evidence for predictability)	R2 (Provides evidence for inner consistence of the design model)	–	–
11c	Formal design and refinement methods	R1 An application-friendly or domain specific specification method and notation	R1 Provides precise definition of limited aspects of behaviour which needs to be appropriate to the domain	R3 Can detect internal inconsistency or missing behaviour or mathematically inconsistent expressions	– Note: May complicate the achievement of this property.	R2 Provides proof for predictability	R2	–	–
11d	Automatic software generation	R1 If executable software is automatically generated from requirements specification, or from a design that has been shown to be complete R2 If the generation tools are shown to have appropriate pedigree	R1 If executable software is automatically generated from requirements specification, or from a design that has been shown to be correct R2 If the generation tools are shown to have appropriate pedigree	R1 If the generation tools guarantee avoidance of particular intrinsic design faults R2 If the generation tools are shown to have appropriate pedigree	–	–	–	R1 R2 R3 If fault tolerance capabilities are automatically generated	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Measure		Properties							
		Completeness with respect to software safety requirements specification	**Correctness with respect to software safety requirements specification**	**Freedom from intrinsic design faults**	**Simplicity and understandability**	**Predictability of behaviour**	**Verifiable and testable design**	**Fault tolerance**	**Defence against common cause failure from external events**
12	Computer-aided specification and design tools	R1 Encapsulation of domain knowledge of the EUC and of the software environment R2 If checklist of issues to be taken into consideration are defined, justified and covered	R1 Enforcement of backward requirements traceability Functional simulation techniques R2 Functional simulation according to defined and justified coverage criteria	R2 Semantic and syntactic checks to ensure that the relevant rules are satisfied	R1 Animation and browsing	–	R2 Semantic and syntactic checks to ensure that the relevant rules are satisfied	–	–
13a	Cyclic behaviour, with guaranteed maximum cycle time	–	R1 for timing aspects of specification R3 If maximum cycle time established by rigorous reasoning	R1 for timing aspects of specification R3 If maximum cycle time established by rigorous reasoning	–	R1 for timing aspects of specification R3 If maximum cycle time established by rigorous reasoning	R1 for timing aspects of specification R3 If maximum cycle time established by rigorous reasoning	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Measure		Properties							
		Completeness with respect to software safety requirements specification	**Correctness with respect to software safety requirements specification**	**Freedom from intrinsic design faults**	**Simplicity and understandability**	**Predictability of behaviour**	**Verifiable and testable design**	**Fault tolerance**	**Defence against common cause failure from external events**
13b	Time-triggered architecture	R3 Completeness is guaranteed by allocation (only for timing properties)	R3 Correctness is guaranteed by allocation (only for timing properties)	R3 Rigorous guarantee against intrinsic timing faults	R1 Defined notation reduces misunderstanding considerably, predictability as approach	R3 Adverse interference: total separation in time domain, no interference	R3 Greatly reduces the effort required for testing and certifying the system	R2 Transparent implementation of fault-tolerance	R3 External interrupts cannot interfere with the time-triggered schedule which gives priority to safety-critical tasks
13c	Event-driven, with guaranteed maximum response time	–	–	–	R1 Event driven architectures may hinder understandibility	R2 Event driven architectures may hinder understandibilit y	R1 Makes testing more predictable	–	–
14	Static resource allocation	R1	R1	R1	R1 Makes the design more understandable	R2 With architecture defining resource usage	R1 Makes testing more predictable	–	–
15	Static synchronisation of access to shared resources	–	R1 Gives predictability in resource access	R1 R3 if supported by rigorous reasoning as to correctness of synchronisation	R1 Makes the design more understandable	R1 R3 if supported by rigorous reasoning as to correctness of synchronisation	–	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Table C.3 – Properties for systematic safety integrity – Software design and development – support tools and programming language

(See 7.4.4. Referenced by Table A.3)

Technique/Measure		Properties		
		Support the production of software with the required software properties	**Clarity of the operation and functionality of the tool**	**Correctness and repeatability of output**
1	Suitable programming language	R2 if strong typing, restricted type conversion. R3 if defined semantics for rigorous reasoning	–	–
2	Strongly typed programming language	R2	–	–
3	Language subset	R2 Depending on chosen subset	R1	R2 Depending on chosen subset
4a	Certificated tools	–	R2	R2
4b	Tools: increased confidence from use	R1 If the class of detected program errors is systematically defined R2 If there is objective validation evidence for the tool performance.	R1 If the tool support is non-specific to the problem domain. R2 If the tool support is significantly specialized to the problem domain.	R1 R2 If there is objective validation evidence for the tool performance e.g. a complier validation suite.

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Table C.4 – Properties for systematic safety integrity – Software design and development – detailed design (includes software system design, software module design and coding)

(See 7.4.5 and 7.4.6. Referenced by Table A.4)

Technique/Measure		Properties							
		Completeness with respect to software safety requirements specification	**Correctness with respect to software safety requirements specification**	**Freedom from intrinsic design faults**	**Simplicity and understandability**	**Predictability of behaviour**	**Verifiable and testable design**	**Fault tolerance / Fault detection**	**Freedom from common cause failure**
1a	Structured methods	R2	R1	R1	–	–	R1 Structured designs are more readily verifiable and testable	–	–
1b	Semi-formal methods	R2	R2	R2	–	R2	R2	–	–
1c	Formal design and refinement methods	–	R3	R3	– Note: May complicate the achievement of this property.	R3 Provides evidence for predictability	R2	–	–
2	Computer-aided design tools	R2 Dependent upon the Computer aided specification tool applying semantic and syntactic checks to ensure that the relevant rules are satisfied	R1	R2 Dependent upon the computer aided specification tool applying semantic and syntactic checks to ensure that the relevant rules are satisfied	–	–	R2 Dependant upon CASE tool to support test coverage and static verification	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Measure		Properties							
		Completeness with respect to software safety requirements specification	**Correctness with respect to software safety requirements specification**	**Freedom from intrinsic design faults**	**Simplicity and understandability**	**Predictability of behaviour**	**Verifiable and testable design**	**Fault tolerance / Fault detection**	**Freedom from common cause failure**
3	Defensive programming	–	–	–	– Note: May complicate the achievement of this property.	– Note: May complicate the achievement of this property.	–	R1 (R2 if coverage targets are defined, justified and met)	R1 (R2 if coverage targets are defined, justified and met)
4	Modular approach	–	–	R1	R1	R1	R1	–	–
5	Design and coding standards	–	–	R1	R1	R1	R1	–	–
6	Structured programming	–	R1	R1	R1	R1	R1	–	–
7	Use of trusted/verified software modules and elements (if available)	–	–	R1 Reuses proven elements	R1 Modular approach decomposes overall complexity into understandable units	R1 The behaviour of the element is already known	–	–	–
8	Forward traceability between the software safety requirements specification and software design	R1 Confidence that the design addresses the software safety requirements	–	–	–	–	–	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Table C.5 – Properties for systematic safety integrity – Software design and development – software module testing and integration

(See 7.4.7 and 7.4.8. Referenced by Table A.5)

Technique/Measure		Properties			
		Completeness of testing and integration with respect to the software design specification	**Correctness of testing and integration with respect to the software design specification (successful completion)**	**Repeatability**	**Precisely defined testing configuration**
1	Probabilistic testing	R1 (R2 if operational profile coverage targets are defined, justified and met)	R1 (R2 if required outputs are defined, justified and met)	–	–
2	Dynamic analysis and testing	R1 (R2 if structural coverage targets are defined, justified and met)	R1 (R2 if required outputs are defined, justified and met)	–	–
3	Data recording and analysis	–	R1	R1 Promotes consistency in testing procedures	R2 If fault records/test logs include details of software baseline
4	Functional and black box testing	R1 (R2 if operational profile coverage targets are defined, justified and met)	R1 (R2 if required outputs are defined, justified and met)	–	–
5	Performance testing	–	R1 (R2 if required outputs are defined, justified and met)	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Measure		Properties			
		Completeness of testing and integration with respect to the software design specification	**Correctness of testing and integration with respect to the software design specification (successful completion)**	**Repeatability**	**Precisely defined testing configuration**
6	Model based testing (MBT)	R2 MBT allows early exposure of ambiguities in specification and design, the MBT process starts with requirements R3 If rigorous reasoning is applied to modelling, and test case generation (TCG) is used	R2 Evaluation of results and regression test suites is a key benefit of MBT R3 If rigorous modelling approach is applied, then objective evidence of coverage is possible	R3 MBT (with TCG) aims at automatic execution of generated tests	R2 MBT is automated, testing configuration has to be precisely defined; execution of the generated tests is similar to black box testing with the possibility to be combined with source code level coverage measurement
7	Interface testing	–	R1 (R2 if required outputs are defined, justified and met)	–	–
8	Test management and automation tools	R1 (R2 if test coverage targets are defined, justified and met)	–	R1 Automation promotes consistency	R2 Gives repeatability of testing
9	Forward traceability between the software safety requirements specification and the module and integration test specifications	R1 Confidence that the test specification addresses the software safety requirements	–	–	R2 Confidence in a clear baseline of requirements under test
10	Formal verification	R3 If rigorous reasoning is applied to construction of test cases to show that all aspects of design have been exercised	R3 Gives objective evidence of meeting all of the software safety requirements	R1 If support tools unavailable R2 If tool supported	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

---,,,,,,,,,,-,,,,,,,,,,.....,,,,,,--

Table C.6 – Properties for systematic safety integrity – Programmable electronics integration (hardware and software)

(See 7.5. Referenced by Table A.6)

	Technique/Measure	Properties			
		Completeness of integration with respect to the design specifications	**Correctness of integration with respect to the design specifications (successful completion)**	**Repeatability**	**Precisely defined integration configuration**
1	Functional and black box testing	R1 (R2 if operational profile coverage targets are defined, justified and met)	R1 (R2 if required outputs are defined, justified and met)	–	–
2	Performance testing	–	R1 (R2 if required outputs are defined, justified and met)	–	–
3	Forward traceability between the system and software design requirements for hardware/software integration and the hardware/software integration test specifications	R1 Confidence that the hardware/software integration test specifications addresses the integration requirements	–	–	R2 Confidence with a clear baseline of requirements under test

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Table C.7 – Properties for systematic safety integrity – Software aspects of system safety validation

(See 7.7. Referenced by Table A.7)

	Technique/Measure	Properties			
		Completeness of validation with respect to the software Design Specification	**Correctness of validation with respect to the software Design Specification (successful completion)**	**Repeatability**	**Precisely defined validation configuration**
1	Probabilistic testing	R1 (R2 if operational profile coverage targets are defined, justified and met)	R1 (R2 if required outputs are defined, justified and met)	–	–
2	Process simulation	R1	R1 (R2 if required outputs are defined, justified and met)	–	R2 Gives a definition of the external environment
3	Functional and black-box testing	R1 (R2 if operational profile coverage targets are defined, justified and met)	R1 (R2 if required outputs are defined, justified and met)	–	–
4	Forward traceability between the software safety requirements specification and the software safety validation plan	R1 Confidence that the software safety validation plan addresses the software safety requirements	–	–	R2 Confidence with a clear baseline of requirements under test
5	Backward traceability between the software safety validation plan and the software safety requirements specification	–	R1 Confidence that software safety validation plan contains no unnecessary complexity	–	R2 Confidence with a clear baseline of requirements under test

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Table C.8 – Properties for systematic safety integrity – Software modification

(See 7.8. Referenced by Table A.8)

	Technique/Measure	Properties					
		Completeness of modification with respect to its requirements	**Correctness of modification with respect to its requirements**	**Freedom from introduction of intrinsic design faults**	**Avoidance of unwanted behaviour**	**Verifiable and testable design**	**Regression testing and verification coverage**
1	Impact analysis	_	_	_	R1	R1	R1
2	Re-verify changed software module	R1 (R2 if objective verification targets)	R1 (R2 if objective verification targets)	R1 (R2 if objective verification targets)	_	_	R1R2 (R2 if objective verification targets)
3	Re-verify affected software modules	R1 (R2 if objective verification targets)	R1 (R2 if objective verification targets)	R1 (R2 if objective verification targets)	_	_	R1 (R2 if objective verification targets)
4a	Revalidate complete system	R1 (R2 if objective verification targets)	R1 (R2 if objective verification targets)	_	R1 (R2 if objective verification targets)	_	R1 (R2 if objective verification targets)
4b	Regression validation	R1 (R2 if objective verification targets)	R1 (R2 if objective verification targets)	_	R1 (R2 if objective verification targets)	_	R1 (R2 if objective verification targets)
5	Software configuration management	_	_	_	_	_	R1
6	Data recording and analysis	R1	R1	_	_	_	_

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

	Technique/Measure	Properties					
		Completeness of modification with respect to its requirements	**Correctness of modification with respect to its requirements**	**Freedom from introduction of intrinsic design faults**	**Avoidance of unwanted behaviour**	**Verifiable and testable design**	**Regression testing and verification coverage**
7	Forward traceability between the software safety requirements specification and the software modification plan (including reverification and revalidation)	R1 Confidence that the software modification plan (including re-verification and revalidation) addresses the software safety requirements	–	–	–	–	–
8	Backward traceability between the software modification plan (including reverification and revalidation) and the software safety requirements specification	–	R1 Confidence that software modification plan (including re-verification and revalidation) contains no unnecessary complexity	–	–	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Table C.9 – Properties for systematic safety integrity – Software verification

(See 7.9. Referenced by Table A.9)

Technique/Measure		Properties			
		Completeness of verification with respect to the previous phase	**Correctness of verification with respect to the previous phase (successful completion)**	**Repeatability**	**Precisely defined verification configuration**
1	Formal proof	–	R3	–	–
2	Animation of specification and design	R1	R1	–	–
3	Static analysis	–	R1/R2/R3 (Rigour may range from language subset enforcement to mathematical formal analysis)	–	–
4	Dynamic analysis and testing	R1 (R2 if structural coverage targets are defined, justified and met)	R1 (R2 if required outputs are defined, justified and met)	–	–
5	Forward traceability between the software Design Specification and the software verification (including data verification) plan.	R1 Confidence that the software verification (including data verification) plan addresses the software safety requirements	–	–	R2 Confidence with a clear baseline of requirements under test
6	Backward traceability between the software verification (including data verification) plan and the software design specification	–	R1 Confidence that software verification (including data verification) plan contains no unnecessary complexity	–	R2 Confidence with a clear baseline of requirements under test
7	Offline numerical analysis	–	R1 Increased confidence in the expected numerical accuracy of well-conditioned calculations (R2 with objective acceptance criteria. R3 if used in conjunction with objective systematic reasoning to justify the acceptance criteria)	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Table C.10 – Properties for systematic safety integrity – Functional safety assessment

(See Clause 8. Referenced by Table A.10)

Technique/Measure		Properties						
		Completeness of functional safety assessment with respect to this standard	**Correctness of functional safety assessment with respect to the design specifications (successful completion)**	**Traceable closure of all identified issues**	**The ability to modify the functional safety assessment after change without the need for extensive re-work of the assessment**	**Repeatability**	**Timeliness**	**Precisely defined configuration**
1	Checklists	R1	R1	R1	–	R1	–	–
2	Decision/truth tables	R1	R2	–	–	R2	–	–
3	Failure analysis	R2	R2	R1 (The failure analysis is based on agreed failure lists)	–	R1 (The failure analysis is based on agreed failure lists)	–	–
4	Common cause failure analysis of diverse software (if diverse software is actually used)	R2	R2	R1 (Provided the CCF analysis is based on agreed CC initiator lists)	–	R1 (Provided the CCF analysis is based on agreed CC initiator lists)	–	–
5	Reliability block diagram	R1	R1	–	–	–	–	–
6	Forward traceability between the requirements of IEC 61508-3 Clause 8 and the plan for software functional safety assessment	R1 Confidence that the plan for software functional safety assessment addresses the requirements of 61508-3 Clause 8	–	–	–	–	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

C.3 Properties for systematic safety integrity – Detailed tables

Table C.11 – Detailed properties – Design and coding standards

(Referenced by Table B.1)

Technique/Measure		Properties							
		Completeness with respect to software safety requirements specification	**Correctness with respect to software safety requirements specification**	**Freedom from intrinsic design faults**	**Simplicity and understandability**	**Predictability of behaviour**	**Verifiable and testable design**	**Fault tolerance / Fault detection**	**Freedom from common cause failure**
1	Use of coding standard to reduce likelihood of errors	–	–	R1	R1 Eliminates selected language constructs	R1	R1	–	–
2	No dynamic objects	–	–	R1/R2/R3 Depending on language used	–	R1/R2/R3 Depending on language used	R1/R2 Depending on language used	–	–
3a	No dynamic variables	–	–	R1/R2/R3 Depending on language used	–	R1/R2/R3 Depending on language used	R1/R2 Depending on language used	–	–
3b	Online checking of the installation of dynamic variables	–	–	R1/R2/R3 Depending on language used	–	R1/R2/R3 Depending on language used	R1/R2 Depending on language used	–	–
4	Limited use of interrupts	–	–	R1/R2 Depending on language used	R1 Increases clarity of logic and event sequences	R1/R2 Depending on language used	R1/R2 Depending on language used	–	–
5	Limited use of pointers	–	–	R1/R2 Depending on language used	R1 Increases clarity of logic	R1/R2 Depending on language used	R1/R2 Depending on language used	–	–
6	Limited use of recursion	–	–	R1/R2 Depending on language used	–	R1/R2 Depending on language used	R1/R2 Depending on language used	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Technique/Measure		Properties							
		Completeness with respect to software safety requirements specification	Correctness with respect to software safety requirements specification	Freedom from intrinsic design faults	Simplicity and understandability	Predictability of behaviour	Verifiable and testable design	Fault tolerance / Fault detection	Freedom from common cause failure
7	No unstructured control flow in programs in higher level languages	–	–	R1/R2 Depending on language used	R1 Increases clarity of logic	R1/R2 Depending on language used	R1/R2 Depending on language used	–	–
8	No automatic type conversion	–	R2 Prevents rounding errors	R2 Prevents rounding errors	R1	R1	–	–	–

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Table C.12 – Detailed properties – Dynamic analysis and testing

(Referenced by Table B.2)

	Technique/Measure	Properties			
		Completeness of testing and verification with respect to the software design specifications	**Correctness of testing and verification with respect to the software design specifications (successful completion)**	**Repeatability**	**Precisely defined testing and verification configuration**
1	Test case execution from boundary value analysis	–	R1 (R2 if objective criteria for boundary results)	–	–
2	Test case execution from error guessing	–	R1	–	–
3	Test case execution from error seeding	–	R1	–	–
4	Test case execution from model-based test case generation	R2 The MBT process starts with requirements and facilitates early finding of errors during software design and development R3 If rigorous reasoning is applied to modelling, and TCG (Test Case Generation) is used	R2 Evaluation of results and regression test suites is a key benefit of MBT, it further facilitates understanding of consequences of specified requirements R3 If rigorous modelling approach is applied, then objective evidence of coverage is possible	R3 MBT (with TCG) aims at automatic execution of generated tests	R2 MBT is automated, testing configuration has to be precisely defined; execution of the generated tests is similar to black box testing with the possibility to be combined with source code level coverage measurement
5	Performance modelling	–	R1 (R2 if objective performance requirements)	–	–
6	Equivalence classes and input partition testing	R1 (If the input data profile is well defined and is manageably simple in structure)	R1 (If the partitions plausibly contain no non-linearities i.e. all members of a class are truly equivalent)	–	–
7	Structure-based testing	–	R1 (R2 is objective structural coverage targets)	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Table C.13 – Detailed properties – Functional and black-box testing

(Referenced by Table B.3)

Technique/Measure		Properties			
		Completeness of testing, integration and validation with respect to the design specifications	**Correctness of testing, integration and validation with respect to the design specifications (successful completion)**	**Repeatability**	**Precisely defined testing, integration and validation configuration**
1	Test case execution from cause consequence diagrams	R1	R1	–	–
2	Test case execution from model-based test case generation	R2 MBT Model-based Testing is the automatic generation of efficient test cases/procedures using models of system requirements and specified functionality, it facilitates early error disclosure and understanding of consequences of specified requirements R3 If rigorous reasoning is applied to modelling, and TCG is used	R2 MBT is based on system models derived from (mainly functional/behavioural) requirements. R3 If rigorous modelling approach is applied, then objective evidence of coverage is possible	R3 MBT (with TCG) aims at automatic execution of generated tests	R2 MBT is automated, testing configuration has to be precisely defined
3	Prototyping/animation	–	R1	–	–
4	Equivalence classes and input partition testing, including boundary value analysis	R1 (If the input data profile is well defined and is manageably simple in structure)	R1 (If the partitions plausibly contain no non-linearities i.e. all members of a class are truly equivalent)	–	–
5	Process simulation	–	R1	–	R2 Gives a definition of the external environment

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Table C.14 – Detailed properties – Failure analysis

(Referenced by Table B.4)

Technique/Measure		Properties						
		Completeness of functional safety assessment with respect to this standard	**Correctness of functional safety assessment with respect to the design specifications (successful completion)**	**Traceable closure of all identified issues**	**The ability to modify the functional safety assessment after change without the need for extensive re-work of the assessment**	**Repeatability**	**Timeliness**	**Precisely defined configuration**
1a	Cause consequence diagrams	R2	R2	–	–	–	–	–
1b	Event tree analysis	R2	R2	–	–	–	–	–
2	Fault tree analysis	R2	R2	–	–	–	–	–
3	Software functional failure analysis	R2	R2	–	–	–	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table C.15 – Detailed properties – Modelling

(Referenced by Table B.5)

	Technique/Measure	Properties			
		Completeness of validation with respect to the software design specification	Correctness of validation with respect to the software design specification (successful completion)	Repeatability	Precisely defined validation configuration
1	Data flow diagrams	–	R1	–	–
2a	Finite state machines	R3	R3	–	–
2b	Formal methods	R3	R3	–	–
2c	Time Petri nets	–	R1	–	–
3	Performance modelling	–	R1	–	–
4	Prototyping/animation	–	R1	–	–
5	Structure diagrams	–	R1	–	–

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Table C.16 – Detailed properties – Performance testing

(Referenced by Table B.6)

Technique/Measure		Properties			
		Completeness of testing and integration with respect to the design specifications	**Correctness of testing and integration with respect to the design specifications (successful completion)**	**Repeatability**	**Precisely defined testing and integration configuration**
1	Avalanche/stress testing	–	R1 (R2 if objective targets are set)	–	–
2	Response timings and memory constraints	–	R1 (R2 if objective targets are set)	–	–
3	Performance requirements	–	R1 (R2 if objective targets are set)	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Table C.17 – Detailed properties – Semi-formal methods

(Referenced by Table B.7)

Technique/Measure		Properties									
		Completeness with respect to software safety requirements specification	**Correctness with respect to software safety requirements specification**	**Freedom from intrinsic design faults**	**Understandable safety requirements**	**Freedom from adverse interference of non-safety functions with the safety needs to be addressed by software**	**Simplicity and understandability**	**Predictability of behaviour**	**Verifiable and testable design**	**Fault tolerance / Fault detection**	**Freedom from common cause failure from external events**
1	Logic/function block diagrams	R2	R2	R2	–	–	R1	R2	–	–	R1
2	Sequence diagrams	R2	R2	R2	–	–	R1	R2	–	–	R2
3	Data flow diagrams	R1	R1	R1	–	–	R1 Suitable for transaction processing	–	–	–	R1
4a	Finite state machines/state transition diagrams	R2	R2	R2	–	–	R1 Mathematically complete specification of event sequences	R2	–	–	R2
4b	Time Petri nets	R2	R2	R2	–	–	R1 Specifies real-time interactions	R2	–	–	R2
5	Entity-relationship-attribute data models	R1	R1	R1	–	–	R1	–	–	–	R1
6	Message sequence charts	R2	R2	R2	–	–	R1	R2	–	–	R2
7	Decision/truth tables	R2	R2	R2	–	–	R1 For combinatorial logic	R2	–	–	R2

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Table C.18 – Properties for systematic safety integrity – Static analysis

(Referenced by Table B.8)

Technique/Measure		Properties			
		Completeness of verification with respect to the previous phase	Correctness of verification with respect to the previous phase (successful completion)	Repeatability	Precisely defined verification configuration
1	Boundary value analysis	–	R1 (R2 if objective criteria for boundary results)	–	–
2	Checklists	–	R1	–	R1
3	Control flow analysis	–	R1	–	–
4	Data flow analysis	–	R1	–	–
5	Error guessing	–	R1	–	–
6a	Formal inspections, including specific criteria	R2	R2	–	R2
6b	Walk-through (software)	R1	R1	–	R1
7	Symbolic execution	–	R2 R3 if used in the context formally defined preconditions and postconditions and performed by a tool using a mathematically rigorous algorithm	–	–
8	Design review	R2	R1 R2 (with objective criteria)	–	R2
9	Static analysis of run time error behaviour	–	R1 R3 for certain classes of error if performed by a tool using a mathematically rigorous algorithm	–	–
10	Worst-case execution time analysis	R1	R3	–	R2

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Table C.19 – Detailed properties – Modular approach

(Referenced by Table B.9)

Technique/Measure		Properties							
		Completeness with respect to software safety requirements specification	Correctness with respect to software safety requirements specification	Freedom from intrinsic design faults	Simplicity and understandability	Predictability of behaviour	Verifiable and testable design	Fault tolerance / Fault detection	Freedom from common cause failure
1	Software module size limit	–	–	R1	R1	R1	R1	–	–
2	Software complexity control	–	–	R1	R1	R1	R1	–	–
3	Information hiding/encapsulation	–	–	R1	R1	R1	R1	–	–
4	Parameter number limit / fixed number of subprogram parameters	–	–	R1	R1	R1	R1	–	–
5	One entry/one exit point in subroutines and functions	–	–	R1	R1	R1	R1	–	–
6	Fully defined interface	–	–	R2	R1	R1	R1	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Annex D
(normative)

Safety manual for compliant items – additional requirements for software elements

D.1 Purpose of the safety manual

D.1.1 When an element is re-used or is intended to be re-used in one or more other system developments, it is necessary to ensure that the element is accompanied by a sufficiently precise and complete description (i.e. functions, constraints and evidence), to make possible an assessment of the integrity of a specific safety function that depends wholly or partly on the element. This shall be implemented by means of a safety manual.

D.1.2 The safety manual may consist of the element supplier's documentation if this is adequate to meet the requirements of Annex D of IEC 61508-2 and of this annex. Otherwise it should be created as part of the design of the safety related system.

D.1.3 The safety manual shall define the attributes of an element, which may comprise hardware constraints and/or software of which the integrator shall be aware and take into consideration during application. In particular it forms the vehicle for informing the integrator of its properties and what the element was designed for, its behaviour and characteristics.

NOTE 1 The scope and time of delivery of the safety manual will be dependent upon who it applies to, the type of integrator, the purpose of the element and who provides and maintains it.

NOTE 2 The person or department or organization that integrates software is called the integrator.

D.2 Contents of the safety manual for a software element

D.2.1 The safety manual shall contain all the information required by IEC 61508-2 Annex D, that is relevant to the element. E.g. the hardware-related items of IEC 61508-2 Annex D are not relevant to a purely software element.

D.2.2 The element shall be identified and all necessary instructions for its use shall be available to the integrator.

NOTE For software this can be demonstrated by clearly identifying the element and demonstrating that its content is unchanged.

D.2.3 Element configuration:

a) The configuration of the software element, the software and hardware run-time environment and if necessary the configuration of the compilation / link system shall be documented in the safety manual.
b) The recommended configuration of the software element shall be documented in the safety manual and that configuration shall be used in safety application.
c) The safety manual shall include all the assumptions made on which the justification for use of the element depends.

D.2.4 The following shall be included in the safety manual:

a) Competence: The minimum degree of knowledge expected of the integrator of the element should be specified, i.e. knowledge of specific application tools.
b) Degree of reliance placed on the element: Details of any certification of the element, independent assessment performed, integrity to which the integrator may place on the

--`,``,,`````````,,``,`,,``,`,`-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

pre-existing element. This should include the integrity to which the element was designed, the standards that were followed during the design process, and any constraints passed to the integrator which shall be implemented in support of the systematic capability claimed. (depending on the functionality of the element, it is conceivable that some requirements may only be met at the integration phase of a system. In such circumstances, these requirements shall be identified for further progression by the integrator. Requirements pertaining to response times and performance are two such examples).

NOTE Unlike IEC 61508-2, IEC 61508-3 does not require software failure modes or quantitative failure rates in safety manual for compliant items, because the causes of software errors are fundamentally different from the causes of the random hardware failures of interest in IEC 61508-2 Annex D.

c) Installation instructions: Details of, or reference to, how to install the pre-existing element into the integrated system.

d) The reason for release of the element: Details of whether the pre-existing element has been subject to release to clear outstanding anomalies, or inclusion of additional functionality.

e) Outstanding anomalies: Details of all outstanding anomalies should be given, with explanation of the anomaly, how it occurs and the mechanisms that the integrator shall take to mitigate the anomaly should the particular functions be used.

f) Backward compatibility: Details of whether the element is compatible with previous releases of the sub-system, and if not, details of the process providing the upgrade path to be followed.

g) Compatibility with other systems: A pre-existing element may be dependent upon a specially developed operating system. In such circumstances, details of the version of the specially developed operating system should be detailed.

The build standard should also be specified incorporating compiler identification and version, tools used in creation of the pre-existing element (identification and version), and test pre-existing element used (again identification and version).

h) Element configuration: Details of the pre-existing element name(s) and description(s) should be given, including the version / issue / modification state.

i) Change control: The mechanism by which the integrator can initiate a change request to the producer of the software.

j) Requirements not met: It is conceivable that there may exist specific requirements that have been specified, but have not been met in the current revision of the element. In such circumstances, these requirements should be identified for the integrator to consider.

k) Design safe state: In certain circumstances, upon controlled failure of the system application, the element may revert to a design safe state. In such circumstances, the precise definition of design safe state should be specified for consideration by the integrator.

l) Interface constraints: Details of any specific constraints, in particular user interface requirements shall be identified.

m) Details of any security measures that may have been implemented against listed threats and vulnerabilities.

n) Configurable elements: details of the configuration method or methods available for the element, their use and any constraints on their use shall be provided.

D.3 Justification of claims in the safety manual for compliant items

D.3.1 All claims in the safety manual for compliant items shall be justified by adequate supporting evidence. See 7.4.9.7 of IEC 61508-2.

NOTE 1 It is essential that the claimed safety performance of an element is supported by sufficient evidence. Unsupported claims do not help establish the correctness and integrity of the safety function to which the element contributes.

--`,,`,,`-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

NOTE 2 The supporting evidence may be derived from the element supplier's own documentation and records of the element supplier's development process, or may be created or supplemented by additional qualification activities by the developer of the safety related system or by third parties.

NOTE 3 There may be commercial or legal restrictions on the availability of the evidence (e.g. copyright or intellectual property rights). These restrictions are outside the scope of this standard.

D.3.2 The supporting evidence that justifies the claims in the safety manual for compliant items is distinct from the element safety manual.

D.3.3 Where the evidence cannot be made available to facilitate functional safety assessment, then the element is not suitable for use in E/E/PE safety-related systems.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Annex E
(informative)

Relationships between IEC 61508-2 and IEC 61508-3

The following table helps finding which clauses of IEC 61508-2 need consideration by those who are dealing with software only and which clauses can be neglected. It is well known that almost all clauses address hardware issues. Therefore this is not repeated here. Important software aspects are treated by IEC 61508-3, many software-related requirements do however also occur in IEC 61508-2, mostly overlapping IEC 61508-3 requirements. Knowledge of IEC 61508-2 is mainly needed for those software specialists who seek compatibility between hardware and software. The IEC 61508-2 requirements are grouped into the following categories:

Table E.1 – Categories of IEC 61508-2 requirements

Software	Both for users of the standard dealing with hardware and for users dealing with software.
Application software	Users dealing with software that is for solving a related safety function as such; not for operating system software or library functions.
System software	For users dealing primarily with operating system software, library functions and the like.
Hardware only	Not for those interested in software only.
Mainly hardware	Concerns software only marginally.

Table E.2 – Requirements of IEC 61508-2 for software and their typical relevance to certain types of software

IEC 61508-2 Requirement	Important to users dealing with	Remarks
7.2	Software	
7.2.3.1	Application software	
7.2.3.2 to 7.2.3.6	Software	
7.2.3.3	Hardware only	
7.3	Software	7.3.2.2 f) Hardware only
7.4	Software	
7.4.2.1 to 7.4.2.12	Software	
7.4.2.13, 7.4.2.14	Hardware only	
7.4.3.1 to 7.4.3.3	Software	
7.4.3.4	Hardware only	
7.4.4	Hardware only	
7.4.5	Hardware only	
7.4.6	Software	7.4.6.7 Hardware only
7.4.7	Software	7.4.7.1 a), b) Hardware only
7.4.8	Hardware only	
7.4.9.1 to 7.4.9.3	Software	
7.4.9.4, 7.4.9.5	Hardware only	
7.4.9.6, 7.4.9.7	Software	
7.4.10	Software	Mainly system software

--`,``,,`````````,,``,`,,``,`,`,`--`,,`,`,`,,`---
Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

IEC 61508-2 Requirement	Important to users dealing with	Remarks
7.4.11	Hardware only	
7.5	Software	
7.6	Software	
7.6.2.1 a)	Hardware	
7.6.2.4	Mainly hardware	
7.7	Software	7.7.2.3, 7.7.2.4 Mainly application software
7.8	Software	
7.9	Mainly Application software	
8	Software	
Annex A.1	Mainly hardware	
Annex A.2 and tables	Mainly hardware	Table A.10 Software
Annex A.3	Mainly hardware	Tables A.16, A.17, A.18 Contain some software aspects
Annex B, all tables	Software	
Annex C	Hardware	
Annex D	Software	D.2.3 Hardware only
Annex E	Hardware only	
Annex F	Hardware only	

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Annex F
(informative)

Techniques for achieving non-interference between software elements on a single computer

F.1 Introduction

Independence of execution between software elements which are hosted on a single computer system (consisting of one or more processors together with memory and other hardware devices shared between those processors) can be achieved and demonstrated by means of a number of different methods. This annex sets out some techniques which can be used to achieve non-interference (between elements of differing systematic capability, between elements which are designed to achieve or contribute to the same safety function, or between software contributing to a safety function and non-safety related software on the same computer).

NOTE The term “independence of execution” means that elements will not adversely interfere with each other’s execution behaviour such that a dangerous failure would occur. It is used to distinguish other aspects of independence which may be required between elements, in particular diversity, to meet other requirements of the standard.

F.2 Domains of behaviour

Independence of execution should be achieved and demonstrated both in the spatial and temporal domains.

Spatial: the data used by a one element shall not be changed by a another element. In particular, it shall not be changed by a non-safety related element.

Temporal: one element shall not cause another element to function incorrectly by taking too high a share of the available processor execution time, or by blocking execution of the other element by locking a shared resource of some kind.

F.3 Causal factor analysis

To demonstrate independence of execution, an analysis of the proposed design should be undertaken to identify all possible causes of execution interference between the notionally independent (non-interfering) elements in the spatial and temporal domains. The analysis should consider both normal operation and operation under failure conditions, and should include (but need not be limited to) the following:

a) shared use of random access memory;

b) shared use of peripheral devices;

c) shared use of processor time (where two or more elements are executed by a single processor);

d) communications between the elements necessary to achieve the overall design;

e) the possibility that a failure in one element (such as an overflow, or divide by zero exception, or an incorrect pointer calculation) may cause a consequent failure in other elements.

The achievement and justification of independence of execution will then have to address all these identified sources of interference.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

F.4 Achieving spatial independence

Techniques for achieving and demonstrating spatial independence include the following:

a) Use of hardware memory protection between different elements, including elements of differing systematic capability.

b) Use of an operating system which permits each element to execute in its own process with its own virtual memory space, supported by hardware memory protection.

c) Use of rigorous design, source code and possibly object code analysis to demonstrate that no explicit or implicit memory references are made from between software elements which can result in data belonging to another element being overwritten (for the case where hardware memory protection is not available).

d) Software protection of the data of a higher integrity element from illegal modification by a lower integrity element.

Data should not be passed from a lower to a higher integrity element unless the higher integrity element can verify that the data is of sufficient integrity.

Where data has to be passed between elements which are required to be independent, uni-directional interfaces such as messages or pipes should be used in preference to shared memory.

NOTE Ideally the independent elements would not communicate with each other. However, where the design of the system requires that one element should send data to another element, the design of the communication mechanism should be such that neither the sending nor the receiving elements should fail or be blocked in execution if data transmission ceases or is delayed.

Any data resident on permanent storage devices such as magnetic discs shall be taken into account for spatial partitioning, in addition to transient data in random access memory. For example, file access protection implemented by an operating system could be used to prevent one element writing to data areas belonging to another element.

F.5 Achieving temporal independence

Techniques for ensuring temporal independence include

a) Deterministic scheduling methods. For example,
   - a cyclic scheduling algorithm which gives each element a defined time slice supported by worst case execution time analysis of each element to demonstrate statically that the timing requirements for each element are met;
   - time triggered architectures.

b) Strict priority based scheduling implemented by a real-time executive with a means of avoiding priority inversion.

c) Time fences which will terminate the execution of an element if it over-runs its allotted execution time or deadline (in such a case, hazard analysis shall be undertaken to show that termination of an element will not result in a dangerous failure, so this technique may be best employed for a non-safety related element).

d) An operating system which guarantees that no process can be starved of processor time, for example by means of time slicing. Such an approach may only be applicable where there are no hard real time requirements to be met by the safety related elements, and it is shown that the scheduling algorithm will not result in undue delays to any element.

Where a resource (such as a peripheral device) is shared between elements, the design shall ensure that the elements will not function incorrectly because the shared resource is locked by another element. The time required to access a shared resource shall be taken into account in determining temporal non-interference.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

F.6 Requirements for supporting software

If an operating system, a real-time executive, memory management, timer management or any other such software is to be used to provide spatial or temporal independence, or both, then such software shall be of the highest systematic capability of any of the elements which are required to be independent.

NOTE It is clear that any such software represents a potential common cause of failure of the independent elements.

F.7 Independence of software modules – programming language aspects

The following Table F.1 is an informal definition of relevant terms.

Table F.1 – Module coupling – definition of terms

Term	Informal definition
Cohesion	measure of tightness of the connections between data and subprograms within one module
Coupling	measure for the tightness of connections between modules
Encapsulation	hiding of internal (private) data and subprograms from external access; term primarily used with object oriented programs
Independence	measure of decoupling of software parts; complement of coupling
Module	confined software part that performs something and that may have data of its own; `Class`, hierarchy of classes, subprogram, `unit`, `module`, `package`, ... according to programming language
Interface	well defined set of heads of subprograms that provide access to a module
Tramp data	data that is not used in the receiving module, but only transferred to another module

As a general rule, module independence is enhanced if there is loose coupling between modules and high cohesion within modules. High cohesion encourages the situation where identifiable units of functionality correspond clearly with identifiable units of implementing code, while loose module coupling promotes low interaction and thus high independence between functionally unrelated modules.

Loose module coupling usually results from achieving high cohesion within modules by putting the code and data together that are used to perform one particular function. Low cohesion results, if code and data are assembled in modules only arbitrarily, or because of some timing sequence or due to some sequence in the control flow.

Several aspects of module coupling can be distinguished, see Table F.2 below.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Table F.2 – Types of module coupling

Coupling	Definition	Explanation	Rationale	Remark
Interface coupling, encapsulation	Coupling only via a well defined set of subprograms.	Access to the module or its data only via subprograms; any change of a value of a variable, any question about the value of such a variable, or any other service required from the module is routed via a subprogram call.	The heads of the subprograms (signatures) of a module explain the available services. If any changes of a module are required, a large amount of these changes can be done within that module, without affecting other modules. Promotes loose coupling, recommended in general.	Mainly for object oriented programs, classes, hierarchies of classes, packages of libraries; not for subprograms.
Data coupling via parameter list	Data transfer only via the parameter list or the identifier of subprograms.	Access to the module or its data only via variables or objects that are indicated in the head of the subprogram; any change of a value of a variable, any question about the value of such a variable is visible.	The head of the subprograms exhibits the data or objects involved with a call of that subprogram. Promotes loose coupling, recommended in general.	Within classes of object oriented programs this principle is normally not observed. Local variables may be accessed directly. Strict adherence to that principle may also lead to tramp data. The principle should be violated to avoid this type of data.
Structure coupling	Data transfer contains more data than necessary.	More data are transferred to the receiving subprogram than necessary for performing the required function.	The superfluous data provide another module with information that it does not require for fulfilling its purpose. These data may lead to misunderstanding the cooperation between the modules. It is, however, not deprecated.	The deficiency can normally easily be corrected.
Control coupling	Coupling that exercises immediate control on the receiving module.	Data transfer that can only cause a branching reaction in the other module; in many cases characterized by transfer of a single bit.	Tighter than the couplings above, as it requires immediate action, prescribing the receiving subprogram to do something. To be handled cautiously; to be avoided, if possible. Not recommended in general.	Cannot always be avoided. May be necessary, e.g. if the completion of an action is announced, or the validity of a value.
Global coupling	Coupling via global data.	Modules can access data that are directly accessible by other modules, or one module can directly access data belonging to another module.	The heads of the subprograms do not indicate, which data are used and from where. It is difficult to understand the subprograms' functions and to predict the effects of any changes to code.	Deprecated in general. May be necessary exceptionally, e.g. to avoid tramp data. To be used only in very limited way that conforms to a clearly defined and documented coding standard.
Content coupling	Jumping directly into other modules, influencing branching goals in other modules, or accessing data in other modules directly.	Feasible in assembly language programs; not possible in all higher level languages. Can accelerate program execution and reduce coding effort.	Deprecated. One module can only be understood by understanding its connected modules as well. Makes a program extremely difficult to understand and extremely difficult to change.	In some programming languages not even possible. Can always be avoided.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Code reading or code review (see 7.9.2.12) should verify whether or not the program modules are loosely coupled. This analysis normally requires some sort of understanding of the modules' purpose and their way of working. Proper coupling can therefore be assessed only by reading the code and its documentation.

Content coupling should be avoided. Global coupling may be used only exceptionally. Control coupling and procedural coupling should be avoided. If ever possible, modules should be connected by interface coupling (encapsulation) and/or data coupling.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Annex G
(informative)

Guidance for tailoring lifecycles associated with data driven systems

G.1 Data driven – system part and application part

Many systems are written in two parts. One part provides the underlying system capability. The other part adapts the system to the specific requirements of the intended application. The application part may be written in the form of data, that configures the system part. This is termed “data driven” in this Annex.

The application specific part of the software, may be developed using a variety of programming tools and programming languages. These languages and tools may constrain the way the application program can be written.

For instance, where a programming language supports the developer/configurer in describing the functionality (e.g. the use of ladder logic for simple interlock systems), then the application software programming task is likely to be fairly simple. However, where the programming language allows the developer/configurer to describe complex application behaviour, then the application software programming task is likely to be complex. Where very simple application software is developed, detailed design may be considered as configuring rather than programming.

The degree of rigour necessary to achieve the required safety integrity is dependent upon the degree of configuration complexity available to the developer/configurer and the complexity of behaviour to be represented in the application. This is represented diagrammatically on the axes of Figure G.1.

For simplicity the axes have been further divided into classes of complexity as:

a) Variability allowed by the language:
   – fixed program;
   – limited variability (some industries view the application program as ‘data’ which is interpreted by the system part);
   – full variability (whilst not normally considered as data driven this type of system may also be used for application development and is included in this annex for completeness).
b) Ability to configure application:
   – limited;
   – full.

In reality a particular system may comprise different levels of complexity and configurability. Further, the complexity may exhibit a sliding scale along the continuum of the two axes. When attempting to tailor the software lifecycle, the relevant level of complexity should be identified and the degree of tailoring should be justified.

A description of the typical types of system for each level of complexity is given below. Guidance on suggested techniques for implementing each type of system is given in IEC 61508-7.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

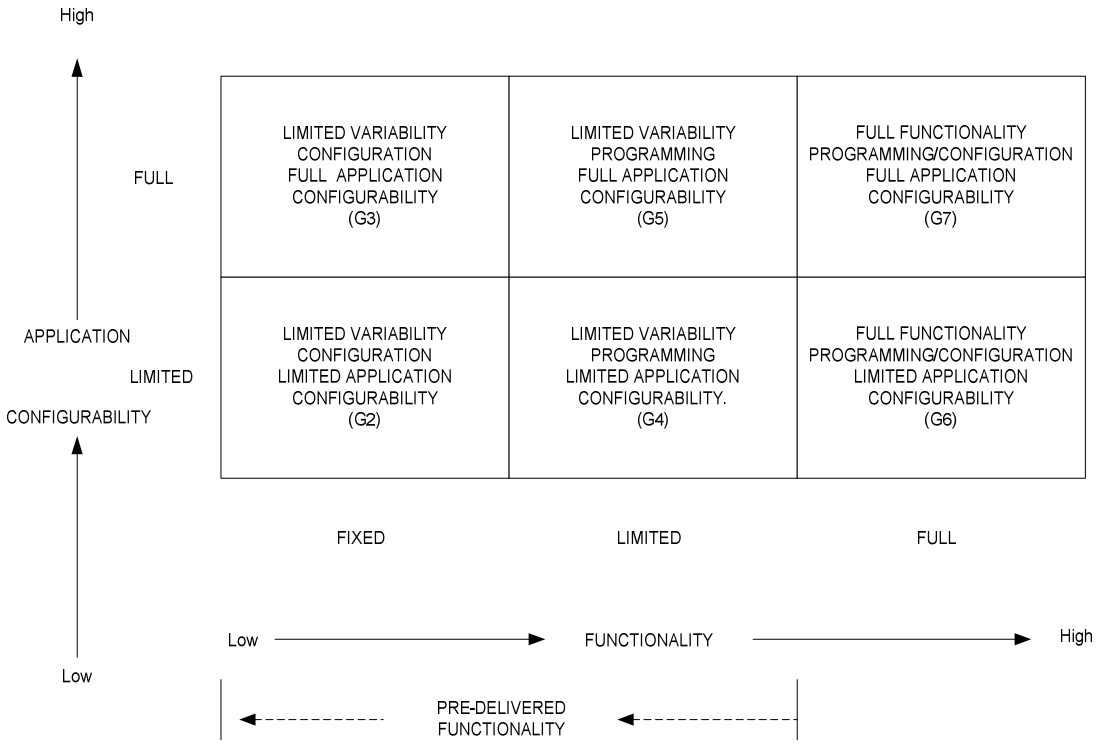


Figure G.1 – Variability in complexity of data driven systems

Typical systems in each class of complexity are described in G.2.

G.2 Limited variability configuration, limited application configurability

A proprietary configuration language used with an IEC 61508 compliant system with fixed pre-delivered functionality.

The configuration language does not allow the programmer to alter the function of the system. Instead configuration is limited to adjustment of a few (data) parameters to enable the system to be matched to its application. Examples may include smart sensors and actuators whereupon specific parameters are entered, network controllers, sequence controllers, small data logging systems and smart instruments.

The justification of the tailoring of the safety lifecycle should include, but not be limited to, the following:

a) specification of the input parameters for this application;
b) verification that the parameters have been correctly implemented in the operational system;
c) validation of all combinations of input parameters;
d) consideration of special and specific modes of operation during configuration;
e) human factors / ergonomics;
f) interlocks, e.g. ensuring that operational interlocks are not invalidated during the configuration process;
g) Inadvertent re-configuration, e.g. key switch access, protection devices.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

G.3 Limited variability configuration, full application configurability

A proprietary configuration language used with an IEC 61508 compliant system with fixed pre-delivered functionality.

The configuration language does not allow the programmer to alter the function of the system. Instead, configuration is constrained to creation of extensive static data parameters to enable the system to be matched to its application. An example may be an air traffic control system consisting of data with large numbers of data entities each with one or more attributes. An essential characteristic of the data is that it contains no explicit sequencing, ordering or branching constructs in the data and does not contain any representation of the combinatorial states of the application.

In addition to the considerations given in G.2, the justification of the tailoring of the safety lifecycle should include, but not be limited to, the following:

a) automation tools for creation of data;

b) consistency checking, e.g. the data is self compatible;

c) rules checking, e.g. to ensure the generation of the data meets the defined constraints;

d) validity of interfaces with the data preparation systems.

G.4 Limited variability programming, limited application configurability

A problem-oriented language, used with an IEC 61508 compliant system, where the language statements contain or resemble the terminology of the application of the user for systems with limited pre-delivered functionality.

These languages allow the user limited flexibility to customize the functions of the system to their own specific requirements, based on a range of hardware and software elements.

An essential characteristic of limited variability programming is that data may contain explicit sequencing, ordering or branching constructs and may invoke combinatorial states of the application. Examples may include functional block programming, ladder logic, spreadsheet based systems, and graphical systems.

In addition to the considerations given in G.3, the following elements should be included, but not limited to:

a) the specification of the application requirements;

b) the permitted language sub-sets for this application;

c) the design methods for combining the language sub-sets;

d) the coverage criteria for verification addressing the combinations of potential system states.

G.5 Limited variability programming, full application configurability

A problem-oriented language, used with an IEC 61508 compliant system, where the language statements contain or resemble the terminology of the application of the user for system with limited pre-delivered functionality.

The essential difference from limited variability programming, limited application configurability is the complexity of the configuration of the application. Examples may include graphical systems and SCADA-based batch control systems.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

In addition to the considerations given in G.4, the following elements should be included but not limited to:

a) the architectural design of the application;

b) the provision of templates;

c) the verification of the individual templates;

d) the verification and validation of the application.

The aspect of the lifecycle outlined in this standard which is most likely to be unnecessary (depending on the language used) is the lowest level module implementation and testing.

G.6 Full functionality programming/configuration, limited application configurability

See G.7 below.

G.7 Full functionality programming/configuration, full application configurability

For these systems the full lifecycle requirements of this standard apply.

Full variability parts of systems are based on general purpose programming languages or general purpose database languages, or general scientific and simulation packages. Typically, these parts will be used in conjunction with a computer-based system, equipped with an operating system which provides system resource allocation and a real time multi-programming environment. Examples of systems that may be written in full variability languages may include for example: a dedicated machinery control system, specially developed flight control systems, or web services for management of safety related services.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Bibliography

[1] IEC 61511 (all parts), *Functional safety – Safety instrumented systems for the process industry sector*

[2] IEC 62061, *Safety of machinery – Functional safety of safety-related electrical, electronic and programmable electronic control systems*

[3] IEC 61800-5-2, *Adjustable speed electrical power drive systems – Part 5-2: Safety requirements – Functional*

[4] 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*

[5] 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*

[6] IEC 61508-7: 2010, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 7: Overview of techniques and measures*

[7] IEC 60601 (all parts), *Medical electrical equipment*

[8] IEC 61131-3, *Programmable controllers – Part 3: Programming languages*

_____

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

SOMMAIRE

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

--`,``,,`````````,,``,`,,``,`,`,`-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

SÉCURITÉ FONCTIONNELLE DES SYSTÈMES ÉLECTRIQUES/ÉLECTRONIQUES/ÉLECTRONIQUES PROGRAMMABLES RELATIFS À LA SÉCURITÉ –

Partie 3: Exigences concernant les logiciels

AVANT-PROPOS

1) La Commission Electrotechnique Internationale (CEI) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI – 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 la CEI"). 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 la CEI, participent également aux travaux. La CEI 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 la CEI concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de la CEI intéressés sont représentés dans chaque comité d'études.
3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI 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 la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de la CEI dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
5) La CEI 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 la CEI. La CEI 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 la CEI, 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 la CEI, 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 la CEI ou de toute autre Publication de la CEI, 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 la CEI peuvent faire l'objet de droits de propriété intellectuelle ou de droits analogues. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de propriété et de ne pas avoir signalé leur existence.

La norme internationale CEI 61508-3 a été établie par le sous-comité 65A: Aspects systèmes, du comité d'études 65 de la CEI: Mesure, commande et automation dans les processus industriels.

Cette deuxième édition annule et remplace la première édition publiée en 1998 dont elle constitue une révision technique.

La présente édition a fait l'objet d'une révision approfondie et intègre de nombreux commentaires reçus lors des différentes phases de révision.

Elle a le statut de publication de sécurité de base conformément au Guide CEI 104.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
65A/550/FDIS	65A/574/RVD

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

Cette publication a été rédigée selon les Directives ISO/CEI, Partie 2.

Une liste de toutes les parties de la série CEI 61508, présentées sous le titre général *Sécurité fonctionnelle des systèmes électriques / électroniques / électroniques programmables relatifs à la sécurité*, peut être consultée sur le site web de la CEI.

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 la CEI sous "http://webstore.iec.ch" dans les données relatives à la publication recherchée. A cette date, la publication sera

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

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

INTRODUCTION

Les systèmes comprenant des composants électriques et/ou électroniques sont utilisés depuis de nombreuses années pour exécuter des fonctions relatives à la sécurité dans la plupart des secteurs d'application. Des systèmes à base d'informatique (dénommés de manière générique systèmes électroniques programmables) sont utilisés dans tous les secteurs d'application pour exécuter des fonctions non relatives à la sécurité, mais aussi de plus en plus souvent relatives à la sécurité. Si l'on veut exploiter efficacement et en toute sécurité la technologie des systèmes informatiques, il est indispensable de fournir à tous les responsables suffisamment d'éléments relatifs à la sécurité pour les guider dans leurs prises de décisions.

La présente Norme internationale présente une approche générique de toutes les activités liées au cycle de vie de sécurité de systèmes électriques et/ou électroniques et/ou électroniques programmables (E/E/PE) qui sont utilisés pour réaliser des fonctions de sécurité. Cette approche unifiée a été adoptée afin de développer une politique technique rationnelle et cohérente concernant tous les systèmes électriques relatifs à la sécurité. Un objectif principal de cette approche est de faciliter le développement de normes internationales de produit et d'application sectorielle basées sur la série CEI 61508.

NOTE 1 Des exemples de normes internationales de produit et d'application sectorielle basées sur la série CEI 61508 sont donnés dans la Bibliographie (voir références [1], [2] et [3]).

Dans la plupart des cas, la sécurité est obtenue par un certain nombre de systèmes fondés sur diverses technologies (par exemple mécanique, hydraulique, pneumatique, électrique, électronique, électronique programmable). En conséquence, toute stratégie de sécurité doit non seulement prendre en compte tous les éléments d'un système individuel (par exemple, les capteurs, les appareils de commande et les actionneurs), mais également prendre en considération tous les systèmes relatifs à la sécurité comme des éléments individuels d'un ensemble complexe. Par conséquent, la présente Norme internationale, bien que traitant des systèmes E/E/PE relatifs à la sécurité, peut aussi fournir un cadre de sécurité susceptible de concerner les systèmes relatifs à la sécurité basés sur d'autres technologies.

Il est admis qu'il existe une grande variété d'applications utilisant des systèmes E/E/PE relatifs à la sécurité dans un grand nombre de secteurs, et couvrant un large éventail de complexité et de potentiel de dangers et de risques. Pour chaque application particulière, les mesures de sécurité requises dépendent de nombreux facteurs propres à l'application. La présente Norme internationale, de par son caractère général, rend désormais possible la prescription de ces mesures dans les futures normes internationales de produit et d'application sectorielle, ainsi que dans les révisions des normes déjà existantes.

La présente Norme internationale

– concerne toutes les phases appropriées du cycle de vie de sécurité global des systèmes E/E/PE et du logiciel (par exemple, depuis le concept initial, en passant par la conception, l'installation, l'exploitation et la maintenance, jusqu'à la mise hors service) lorsque les systèmes E/E/PE permettent d'exécuter des fonctions de sécurité,
– a été élaborée dans le souci de la prise en compte de l'évolution rapide des technologies; le cadre fourni par la présente Norme internationale est suffisamment solide et étendu pour pourvoir aux évolutions futures,
– permet l'élaboration de normes internationales de produit et d'application sectorielle concernant les systèmes E/E/PE relatifs à la sécurité; il convient que l'élaboration de normes internationales de produit et d'application sectorielle dans le cadre de la présente norme, permette d'atteindre un haut niveau de cohérence (par exemple, pour ce qui est des principes sous-jacents, de la terminologie, etc.) à la fois au sein de chaque secteur d'application, et d'un secteur à l'autre. La conséquence en sera une amélioration en termes de sécurité et de gains économiques,
– fournit une méthode de définition d'une spécification des exigences de sécurité nécessaire pour obtenir la sécurité fonctionnelle requise des systèmes E/E/PE relatifs à la sécurité,

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

– adopte une approche basée sur les risques qui permet de déterminer les exigences en matière d'intégrité de sécurité,

– introduit les niveaux d'intégrité de sécurité pour la spécification du niveau cible d'intégrité de sécurité des fonctions de sécurité devant être réalisées par les systèmes E/E/PE relatifs à la sécurité,

NOTE 2 La norme ne spécifie aucune exigence de niveau d'intégrité de sécurité pour aucune fonction de sécurité, ni comment le niveau d'intégrité de sécurité est déterminé. Elle fournit en revanche un cadre conceptuel basé sur les risques, ainsi que des exemples de méthodes.

– fixe des objectifs chiffrés de défaillance pour les fonctions de sécurité exécutées par les systèmes E/E/PE relatifs à la sécurité, qui sont en rapport avec les niveaux d'intégrité de sécurité,

– fixe une limite inférieure pour les objectifs chiffrés de défaillance pour une fonction de sécurité exécutée par un système E/E/PE relatif à la sécurité unique. Pour des systèmes E/E/PE relatifs à la sécurité fonctionnant

  – en mode de fonctionnement à faible sollicitation, la limite inférieure est fixée pour une probabilité moyenne de défaillance dangereuse de 10^{-5} en cas de sollicitation,

  – en mode de fonctionnement continu ou à sollicitation élevée, la limite inférieure est fixée à une fréquence moyenne de défaillance dangereuse de 10^{-9} [h^{-1}],

NOTE 3 Un système E/E/PE relatif à la sécurité unique n'implique pas nécessairement une architecture à un seul canal.

NOTE 4 Dans le cas de systèmes non complexes, il peut être possible de concevoir des systèmes relatifs à la sécurité ayant des valeurs plus basses pour l'intégrité de sécurité cible. Il est toutefois considéré que ces limites représentent ce qui peut être réalisé à l'heure actuelle pour des systèmes relativement complexes (par exemple, des systèmes électroniques programmables relatifs à la sécurité).

– établit des exigences fondées sur l'expérience et le jugement acquis dans le domaine des applications industrielles afin d'éviter des anomalies systématiques ou pour les maintenir sous contrôle. Même si, en général, la probabilité d'occurrence des défaillances systématiques ne peut être quantifiée, la norme permet cependant pour une fonction de sécurité spécifique, de déclarer que l'objectif chiffré de défaillance associé à cette fonction de sécurité peut être réputé atteint si toutes les exigences de la norme sont remplies,

– introduit une capabilité systématique s'appliquant à un élément du fait qu'il permet d'assurer que l'intégrité de sécurité systématique satisfait aux exigences du niveau d'intégrité de sécurité spécifié,

– adopte une large gamme de principes, techniques et mesures pour la réalisation de la sécurité fonctionnelle des systèmes E/E/PE relatifs à la sécurité, mais n'utilise pas de manière explicite le concept de sécurité intrinsèque. Les principes de « sécurité intrinsèque » peuvent toutefois être applicables, l'adoption de ces concepts étant par ailleurs acceptable sous réserve de la satisfaction aux exigences des articles concernés de la norme.

--`,,``,,`````````,,``,,,,``,,`,`-`-`,,,,,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

SÉCURITÉ FONCTIONNELLE DES SYSTÈMES ÉLECTRIQUES/ÉLECTRONIQUES/ÉLECTRONIQUES PROGRAMMABLES RELATIFS À LA SÉCURITÉ –

Partie 3: Exigences concernant les logiciels

1 Domaine d'application

1.1 La présente partie de la série CEI 61508

a) est destinée à n'être utilisée qu'après s'être assuré d'une compréhension parfaite de la CEI 61508-1 et de la CEI 61508-2;

b) s'applique à tout logiciel faisant partie intégrante d'un système relatif à la sécurité ou utilisé pour développer un système relatif à la sécurité entrant dans le domaine d'application de la CEI 61508-1 et de la CEI 61508-2. Ce type de logiciel est désigné par le terme "logiciel de sécurité" (comprenant les systèmes d'exploitation, les logiciels système, les logiciels des réseaux de communication, les fonctions d'interface homme-machine et les micrologiciels, ainsi que les logiciels d'application);

c) fournit des exigences spécifiques applicables aux outils de support utilisés pour développer et configurer un système relatif à la sécurité dans le cadre du domaine d'application de la CEI 61508-1 et de la CEI 61508-2;

d) nécessite que les fonctions de sécurité du logiciel et la capabilité systématique du logiciel soient précisées;

NOTE 1 Si cela a déjà été réalisé dans le cadre de la spécification des systèmes E/E/PE relatifs à la sécurité (voir 7.2 de la CEI 61508-2), il n'est alors pas nécessaire de le répéter dans la présente partie.

NOTE 2 Spécifier les fonctions de sécurité du logiciel et la capabilité systématique du logiciel constitue une procédure itérative; voir les Figures 3 et 6.

NOTE 3 Voir l'Article 5 et l'Annexe A de la CEI 61508-1 pour la structure de la documentation. Cette structure peut tenir compte des procédures de l'entreprise et des pratiques professionnelles de secteurs d'application spécifiques.

e) établit des exigences concernant les phases et activités du cycle de vie de sécurité qui doivent être appliquées durant la conception et le développement du logiciel de sécurité (modèle de cycle de vie de sécurité du logiciel). Ces exigences comprennent l'application de mesures et de techniques qui suivent une gradation basée sur la capabilité systématique requise, afin d'éviter et de maîtriser les anomalies et défaillances du logiciel;

f) fournit les exigences pour les informations relatives aux aspects du logiciel applicables à la validation de la sécurité du système et devant être transmises à l'organisation en charge de l'intégration du système E/E/PE;

g) fournit les exigences pour la préparation des informations et procédures concernant le logiciel requises par l'utilisateur pour le fonctionnement et la maintenance d'un système E/E/PE relatif à la sécurité;

h) fournit les exigences devant être observées par l'organisation en charge des modifications du logiciel de sécurité;

i) fournit, en accord avec la CEI 61508-1 et la CEI 61508-2, les exigences pour les outils de support tels que les outils de conception et développement, les traducteurs de langage, les outils d'essai et de mise au point et les outils de gestion de configuration;

NOTE 4 La Figure 5 montre la relation entre la CEI 61508-2 et la CEI 61508-3.

j) ne s'applique pas aux appareils médicaux conformes à la série CEI 60601.

1.2 Les CEI 61508-1, CEI 61508-2, CEI 61508-3 et CEI 61508-4 sont des publications fondamentales de sécurité, bien que ce statut ne soit pas applicable dans le contexte des

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

systèmes E/E/PE de faible complexité relatifs à la sécurité (voir 3.4.3 de la CEI 61508-4). En tant que publications fondamentales de sécurité, ces normes sont destinées à être utilisées par les comités d'études pour la préparation des normes conformément aux principes contenus dans le Guide CEI 104 et le Guide ISO/CEI 51. Les CEI 61508-1, CEI 61508-2, CEI 61508-3 et CEI 61508-4 sont également destinées à être utilisées comme publications autonomes. La fonction de sécurité horizontale de la présente norme internationale ne s'applique pas aux appareils médicaux conformes à la série CEI 60601.

1.3 Une des responsabilités incombant à un comité d'études consiste, dans toute la mesure du possible, à utiliser les publications fondamentales de sécurité pour la préparation de ses publications. Dans ce contexte, les exigences, les méthodes ou les conditions d'essai de cette publication fondamentale de sécurité ne s'appliquent que si elles sont indiquées spécifiquement ou incluses dans les publications préparées par ces comités d'études.

1.4 La Figure 1 illustre la structure générale de la série CEI 61508 et montre le rôle que la CEI 61508-3 joue dans la réalisation de la sécurité fonctionnelle pour les systèmes E/E/PE relatifs à la sécurité.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

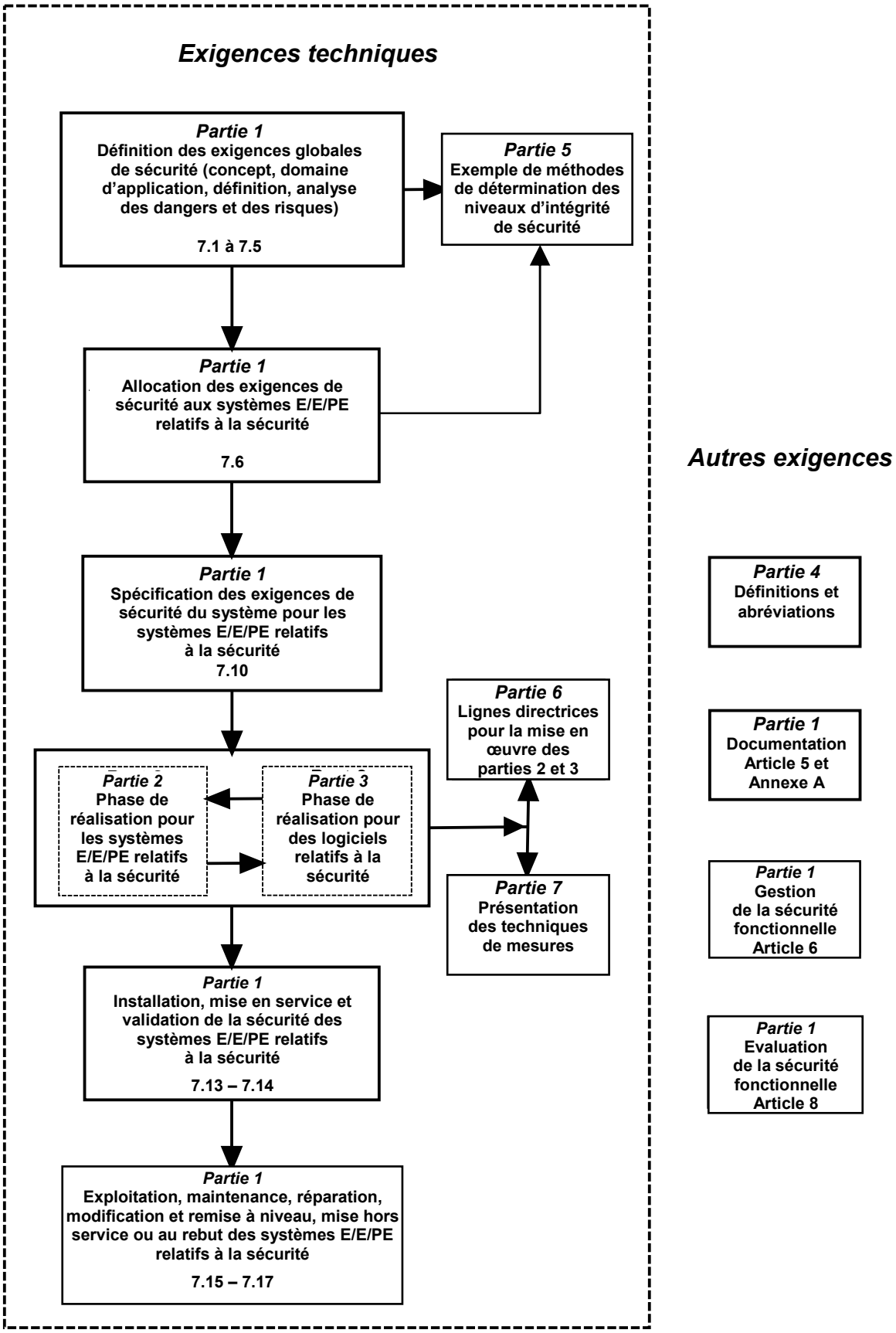


Figure 1 – Structure générale de la série CEI 61508

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

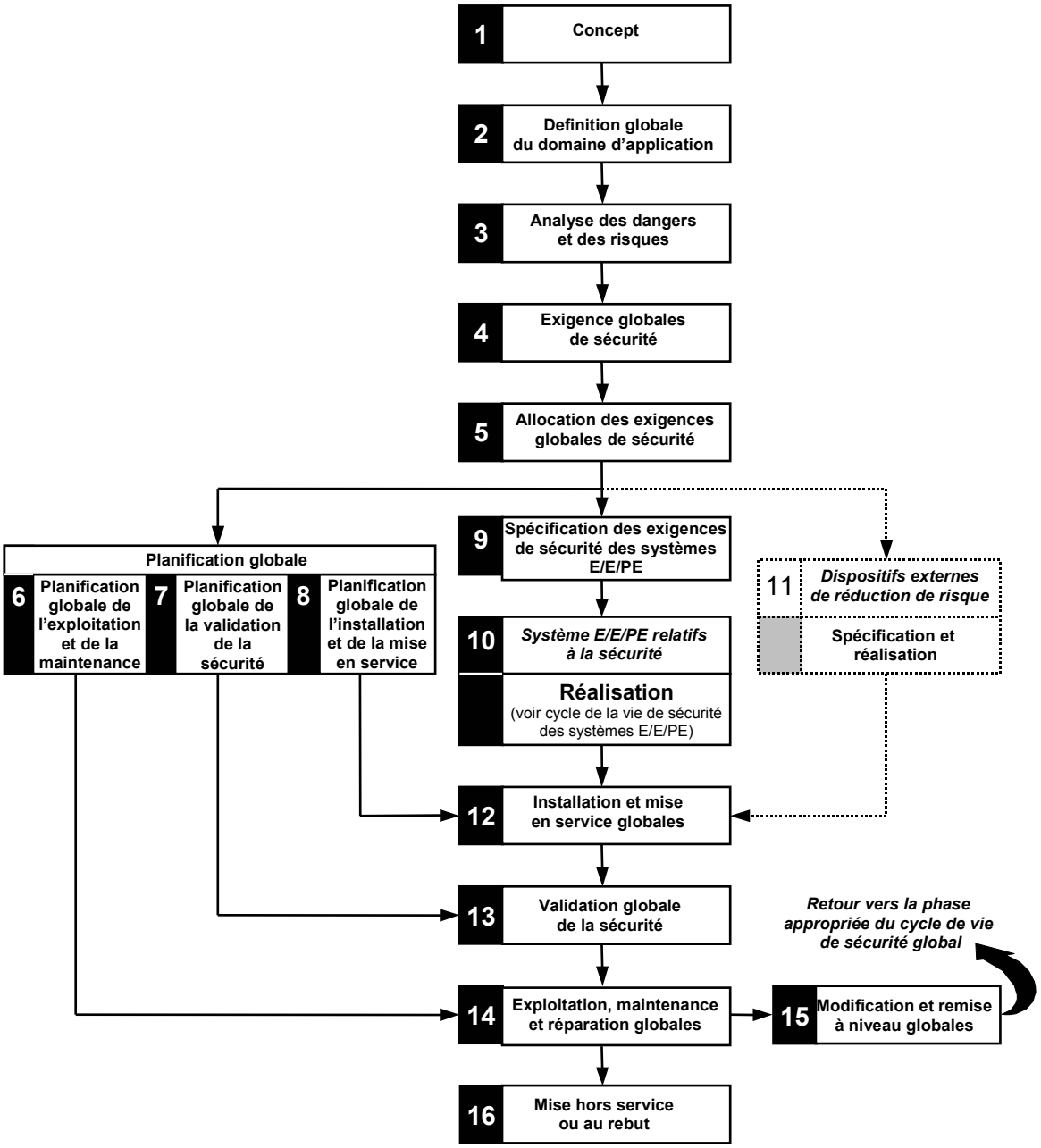


Figure 2 – Cycle de vie de sécurité global

2 Références normatives

Les documents de référence suivants sont indispensables pour l'application 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).

CEI 61508-1: 2010, *Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité – Partie 1: Exigences générales*

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

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

CEI 61508-4: 2010, *Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité – Partie 4: Définitions et abréviations*

Guide CEI 104:1997, *Elaboration des publications de sécurité et utilisation des publications fondamentales de sécurité et publications groupées de sécurité*

Guide ISO/CEI 51:1999, *Aspects liés à la sécurité – Principes directeurs pour les inclure dans les normes*

3 Définitions et abréviations

Pour les besoins du présent document, les définitions et les abréviations données dans la CEI 61508-4 s'appliquent.

4 Conformité à la présente norme

Les exigences de conformité à la présente norme figurent à l'Article 4 de la CEI 61508-1.

5 Documentation

Les objectifs et exigences concernant la documentation figurent à l'Article 5 de la CEI 61508-1.

6 Exigences supplémentaires pour la gestion du logiciel de sécurité

6.1 Objectifs

Les objectifs sont détaillés au 6.1 de la CEI 61508-1.

6.2 Exigences

6.2.1 Les exigences sont celles qui sont détaillées au 6.2 de la CEI 61508-1, ainsi que les exigences supplémentaires suivantes.

6.2.2 La planification de la sécurité fonctionnelle doit définir la stratégie pour l'approvisionnement, le développement, l'intégration, la vérification, la validation et la modification du logiciel dans les limites requises par le niveau d'intégrité de sécurité des fonctions de sécurité mises en œuvre par le système E/E/PE relatif à la sécurité.

NOTE La philosophie de cette approche est de saisir l'opportunité de la planification de la sécurité fonctionnelle pour adapter la présente norme à la prise en compte de l'intégrité de sécurité requise pour chaque fonction de sécurité mise en œuvre par le système E/E/PE relatif à la sécurité.

6.2.3 La gestion de configuration du logiciel doit:

a) maîtriser administrativement et techniquement, tout au long du cycle de vie de sécurité du logiciel, la gestion des modifications du logiciel, assurant ainsi la conformité permanente aux exigences pour le logiciel de sécurité spécifiées;

b) garantir que toutes les opérations nécessaires ont été effectuées en vue de démontrer que la capabilité systématique requise du logiciel a été atteinte;

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

c) maintenir de manière précise et au moyen d'une identification unique tous les éléments de configuration qui sont nécessaires pour satisfaire aux exigences d'intégrité de sécurité du système E/E/PE relatif à la sécurité. La configuration comprend au moins les éléments suivants: analyse de sécurité et exigences, documents de conception et de spécification du logiciel, modules de code source du logiciel, plans d'essai et résultats, documents de vérification, progiciels et composants logiciels préexistants à incorporer au système E/E/PE relatif à la sécurité et tous les outils et environnements de développement qui sont utilisés pour créer, soumettre à l'essai ou effectuer une action sur le logiciel du système E/E/PE relatif à la sécurité;

d) appliquer des procédures de gestion des modifications:

- afin d'empêcher toute modification non autorisée; de documenter les demandes de modification;
- d'analyser l'impact d'une modification proposée, et d'approuver ou rejeter la demande de modification;
- de documenter les détails et les autorisations pour toutes les modifications approuvées;
- d'établir le référentiel de la configuration à des points appropriés lors du développement du logiciel, et de documenter l'essai d'intégration (partielle) du référentiel;
- de garantir la composition et la construction, de tous les référentiels logiciels (y compris la reconstruction de référentiels précédents).

NOTE 1 Une autorité de gestion et de décision est nécessaire pour guider et assurer la maîtrise administrative et technique.

NOTE 2 Une analyse d'impact pouvant comprendre une évaluation informelle constitue un cas extrême. Une analyse d'impact pouvant comprendre une évaluation formelle rigoureuse de l'impact défavorable potentiel de toutes les modifications proposées susceptibles d'être comprises ou mises en œuvre de manière inappropriée, constitue un autre cas extrême. Voir la CEI 61508-7 pour les recommandations concernant l'analyse d'impact.

e) assurer que des méthodes appropriées sont mises en œuvre pour charger correctement des composants logiciels et des données valides dans le système d'exécution;

NOTE 3 Ceci peut comprendre de prévoir des systèmes de localisation cible spécifiques ainsi que des systèmes généraux. Des logiciels autres que ceux relatifs à l'application peuvent nécessiter d'appliquer une méthode de chargement sûre, par exemple, un micrologiciel.

f) documenter les informations suivantes afin de permettre un audit ultérieur: état de la configuration, état des versions, justification (compte tenu de l'analyse d'impact) et approbation de toutes les modifications, et détails de la modification;

g) documenter de manière formelle la version du logiciel de sécurité. Les copies originales du logiciel et toute la documentation associée, ainsi que la version des données en vigueur, doivent être conservées afin de permettre la maintenance et la modification au cours de l'exploitation de la version du logiciel.

NOTE 4 Pour tout renseignement complémentaire concernant la gestion de la configuration, voir la CEI 61508-7.

7 Exigences concernant le cycle de vie de sécurité du logiciel

7.1 Généralités

7.1.1 Objectif

L'objectif des exigences du présent paragraphe est de structurer le développement du logiciel sous forme de phases et d'activités définies (voir Tableau 1 et Figures 3 à 6).

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.1.2 Exigences

7.1.2.1 Un cycle de vie de sécurité pour le développement du logiciel doit être sélectionné et spécifié pendant la planification de sécurité conformément à l'Article 6 de la CEI 61508-1.

7.1.2.2 Tout modèle de cycle de vie du logiciel peut être utilisé à condition que tous les objectifs et exigences du présent article soient remplis.

7.1.2.3 Chaque phase du cycle de vie de sécurité du logiciel doit être divisée en activités élémentaires avec, pour chaque phase, la spécification du domaine d'application, des entrées et des sorties.

NOTE Voir les Figures 3, 4 et le Tableau 1.

7.1.2.4 A condition que le cycle de vie de sécurité du logiciel satisfasse aux exigences du Tableau 1, il est acceptable d'adapter le modèle en V (voir Figure 6), afin de tenir compte de l'intégrité de sécurité et de la complexité du projet.

NOTE 1 Un modèle de cycle de vie de sécurité du logiciel conforme aux exigences du présent article peut être adapté de manière appropriée aux besoins particuliers du projet ou de l'organisation. La liste complète des phases du cycle de vie fournie dans le Tableau 1 convient pour de grands systèmes nouvellement développés. Pour les petits systèmes, il peut être approprié, par exemple, de fusionner les phases de conception du système logiciel et de conception d'architecture.

NOTE 2 Voir l'Annexe G pour les caractéristiques des systèmes dirigés par les données (par exemple, langages de programmation à variabilité totale / variabilité limitée, étendue de la configuration des données) qui peuvent se révéler pertinentes lors de la personnalisation du cycle de vie de sécurité du logiciel.

7.1.2.5 Toute personnalisation du cycle de vie de sécurité du logiciel doit être justifiée sur la base de la sécurité fonctionnelle.

7.1.2.6 Les procédures d'assurance qualité et sécurité doivent être intégrées dans les activités du cycle de vie de sécurité.

7.1.2.7 Pour chaque phase du cycle de vie, des techniques et mesures appropriées doivent être utilisées. Les Annexes A et B donnent des recommandations pour la sélection de techniques et mesures et des références aux CEI 61508-6 et CEI 61508-7. La CEI 61508-6 et la CEI 61508-7 donnent des recommandations sur les techniques spécifiques permettant d'obtenir les propriétés requises pour l'intégrité systématique. De ce fait, sélectionner des techniques issues de ces recommandations ne garantit pas que l'intégrité de sécurité requise est atteinte.

NOTE La réalisation satisfaisante de l'intégrité systématique dépend du choix des techniques en accordant une attention toute particulière aux facteurs suivants:

– la cohérence et la complémentarité des méthodes, langages et outils choisis pour l'ensemble du cycle de développement;

– le fait de déterminer si les développeurs utilisent des méthodes, langages et outils qu'ils comprennent parfaitement;

– le fait de déterminer si les méthodes, langages et outils sont bien adaptés aux problèmes spécifiques rencontrés pendant le développement.

7.1.2.8 Les résultats des activités menées dans le cadre du cycle de vie de sécurité du logiciel doivent être documentés (voir l'Article 5).

NOTE L'Article 5 de la CEI 61508-1 prend en compte les sorties documentées des phases du cycle de vie de sécurité. Durant le développement de certains systèmes E/E/PE relatifs à la sécurité, la sortie de certaines phases du cycle de vie de sécurité peut être couverte par un document distinct tandis que les sorties documentées de plusieurs phases peuvent être fusionnées. L'exigence principale veut que la sortie de la phase du cycle de vie de sécurité soit adaptée à l'objet prévu.

7.1.2.9 Si, à toute phase du cycle de vie de sécurité du logiciel, il est nécessaire d'effectuer une modification portant sur une phase précédente du cycle de vie, une analyse d'impact doit alors déterminer (1) les modules logiciels concernés et (2) les activités précédentes du cycle de vie de sécurité qui doivent être répétées.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

NOTE Une analyse d'impact pouvant comprendre une évaluation informelle constitue un cas extrême. Une analyse d'impact pouvant comprendre une évaluation formelle rigoureuse de l'impact défavorable potentiel de toutes les modifications proposées susceptibles d'être comprises ou mises en œuvre de manière inappropriée, constitue un autre cas extrême. Voir la CEI 61508-7 pour les recommandations concernant l'analyse d'impact.

--`,,`,,`````````,,`,,`,,`,`,,`~-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

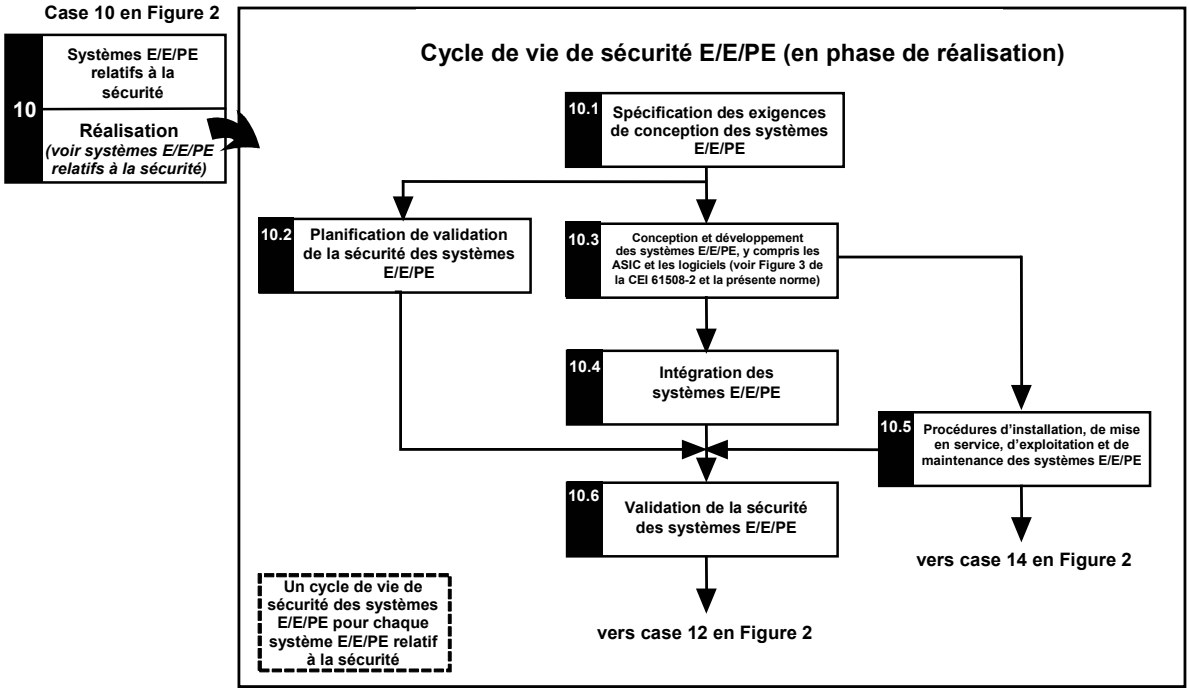


Figure 3 – Cycle de vie de sécurité du système E/E/PE (en phase de réalisation)

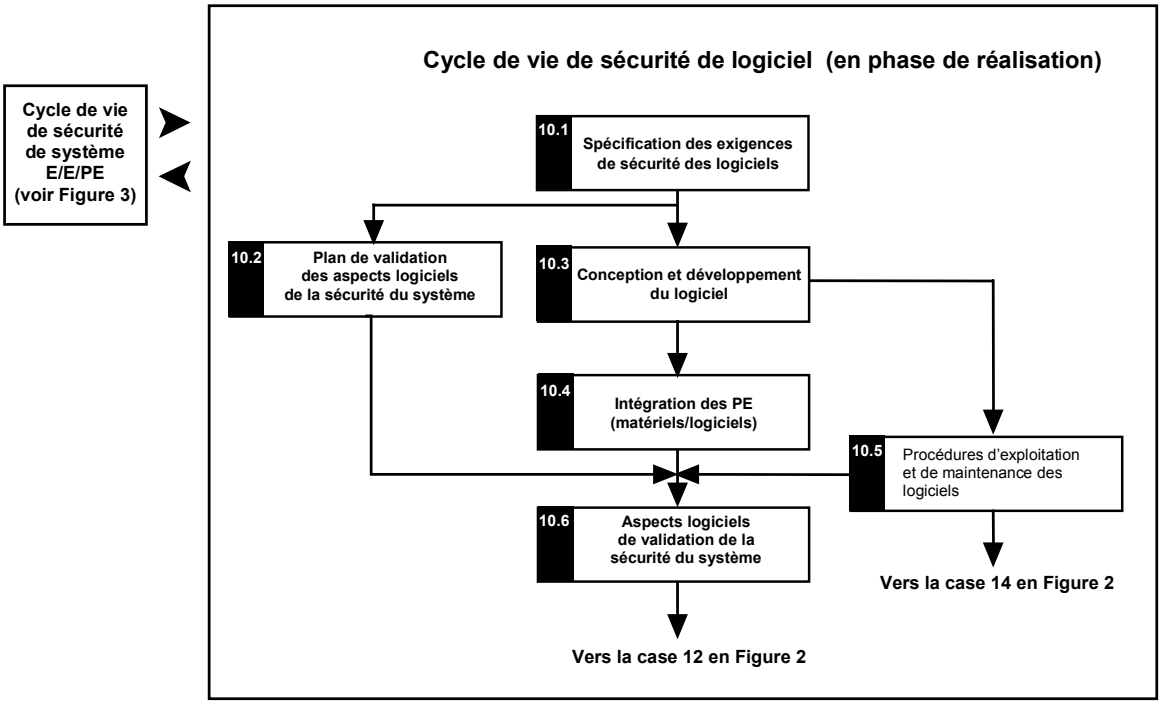


Figure 4 – Cycle de vie de sécurité du logiciel (en phase de réalisation)

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

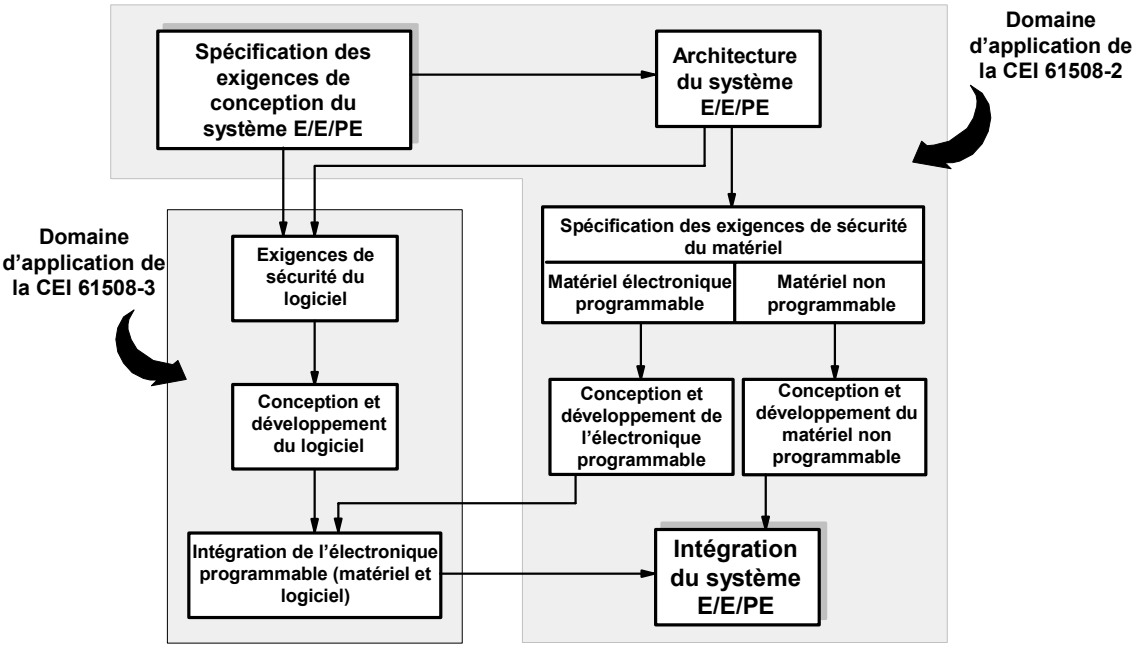


Figure 5 – Relation et domaine d'application pour la CEI 61508-2 et la CEI 61508-3

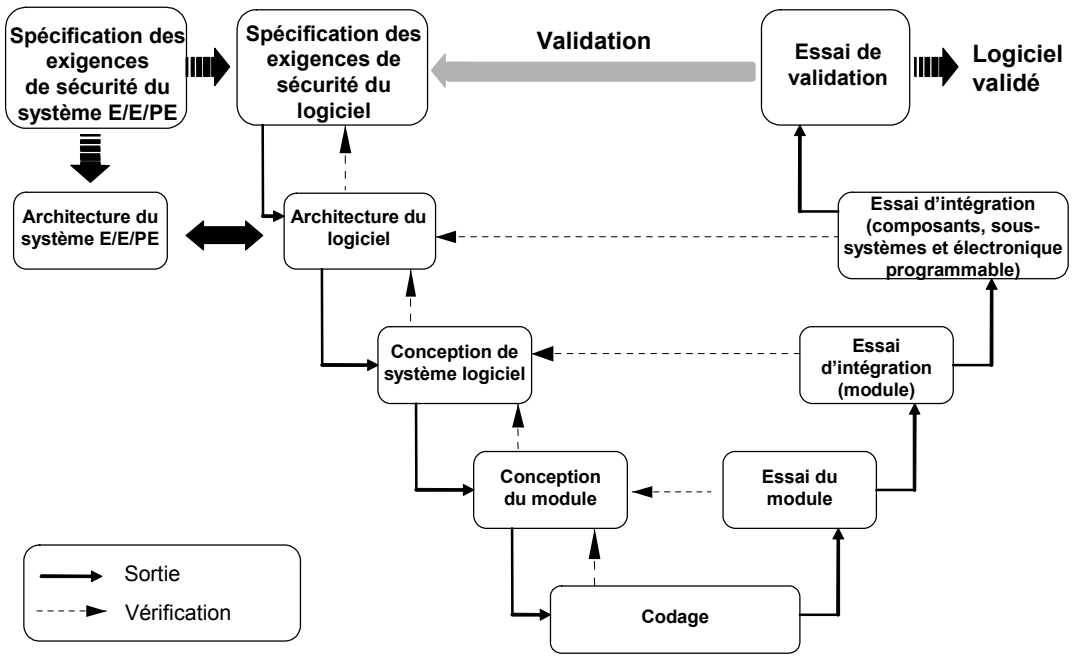


Figure 6 – Capabilité systématique du logiciel et cycle de vie de développement (modèle en V)

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Tableau 1 – Cycle de vie de sécurité du logiciel – présentation

Phase du cycle de vie de sécurité		Objectifs	Domaine d'application	Paragraphe contenant les exigences	Entrées (information requise)	Sorties (information fournie)
Numéro de case de la Figure 4	Titre					
10.1	Spécification des exigences pour la sécurité du logiciel	Spécifier les exigences pour le logiciel de sécurité en termes d'exigences relatives aux fonctions de sécurité du logiciel et d'exigences relatives à la capabilité systématique du logiciel; Spécifier les exigences relatives aux fonctions de sécurité du logiciel pour chaque système E/E/PE relatif à la sécurité nécessaire pour implémenter les fonctions de sécurité requises; Spécifier les exigences relatives à la capabilité systématique du logiciel pour chaque système E/E/PE relatif à la sécurité permettant d'atteindre le niveau d'intégrité de sécurité spécifié pour chaque fonction de sécurité qui lui est attribuée	Système PE; système logiciel	7.2.2	Spécification des exigences de sécurité des E/E/PE développée lors de l'allocation (voir CEI 61508-1) Spécification des exigences de sécurité relatives aux systèmes E/E/PE (issues de la CEI 61508-2)	Spécification des exigences pour la sécurité du logiciel
10.2	Planification de la validation de sécurité du logiciel	Développer un plan de validation de sécurité du logiciel	Système PE; système logiciel	7.3.2	Spécification des exigences pour la sécurité du logiciel	Planification de la validation de sécurité du logiciel
10.3	Conception et développement du logiciel	Architecture: Créer une architecture de logiciel satisfaisant aux exigences spécifiées pour le logiciel de sécurité en fonction du niveau d'intégrité de sécurité requis; Evaluer les exigences imposées au logiciel par l'architecture matérielle du système E/E/PE relatif à la sécurité, y compris les conséquences des interactions matériel/logiciel de l'E/E/PE sur la sécurité de l'équipement commandé	Système PE; système logiciel	7.4.3	Spécification des exigences pour la sécurité du logiciel; Conception de l'architecture matérielle du système E/E/PE (issue de la CEI 61508-2)	Conception de l'architecture du logiciel; Spécification d'essai d'intégration de l'architecture logicielle Spécification d'essai d'intégration du logiciel/système PE (également requis par la CEI 61508-2)

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Tableau 1 *(suite)*

Phase du cycle de vie de sécurité		Objectifs	Domaine d'application	Paragraphe contenant les exigences	Entrées (information requise)	Sorties (information fournie)
Numéro de case de la Figure 4	Titre					
10.3	Conception et développement du logiciel	Outils de support et langages de programmation: Sélectionner un ensemble approprié d'outils d'aide à la vérification, la validation, l'évaluation et la modification, y compris les langages et les compilateurs, les interfaces système d'exécution, les interfaces utilisateurs et les formats et représentations des données pour le niveau d'intégrité de sécurité requis au cours du cycle de vie de sécurité complet du logiciel	Système PE ; système logiciel ; outils de support ; langage de programmation	7.4.4	spécification des exigences pour la sécurité du logiciel ; conception de l'architecture du logiciel	outils de support et règles de codage ; choix des outils de développement
10.3	Conception et développement du logiciel	Conception détaillée et développement (conception du système logiciel): Concevoir et implémenter un logiciel qui répond aux exigences spécifiées pour le logiciel de sécurité en fonction du niveau d'intégrité de sécurité requis, qui soit analysable et vérifiable, et qui soit modifiable de façon sûre	principaux composants et sous-systèmes de la conception de l'architecture du logiciel.	7.4.5	conception de l'architecture du logiciel ; outils de support et règles de codage.	Spécification de la conception du système logiciel ; spécification d'essai d'intégration du système logiciel.
10.3	Conception et développement du logiciel	Conception détaillée et développement (conception des modules logiciels individuels): Concevoir et implémenter un logiciel qui répond aux exigences spécifiées pour le logiciel de sécurité en fonction du niveau d'intégrité de sécurité requis, qui soit analysable et vérifiable, et qui soit modifiable de façon sûre	conception du système logiciel	7.4.5	spécification de la conception du système logiciel ; outils de support et règles de codage	spécification de la conception des modules logiciels ; spécification d'essai des modules logiciels
10.3	conception et développement du logiciel	Implémentation du code détaillé: Concevoir et implémenter un logiciel qui répond aux exigences spécifiées pour le logiciel de sécurité en fonction du niveau d'intégrité de sécurité requis, qui soit analysable et vérifiable, et qui soit modifiable de façon sûre	modules logiciels individuels	7.4.6	spécification de la conception des modules logiciels ; outils de support et règles de codage	liste des codes source ; rapport de revue de code

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Tableau 1 *(suite)*

Phase du cycle de vie de sécurité		Objectifs	Domaine d'application	Paragraphe contenant les exigences	Entrées (information requise)	Sorties (information fournie)
Numéro de case de la Figure 4	Titre					
10.3	Conception et développement du logiciel	Essai des modules logiciels: Vérifier que les exigences pour le logiciel de sécurité (en termes de fonctions de sécurité du logiciel requises et de capabilité systématique du logiciel) ont été remplies. Montrer que chaque module logiciel remplit sa fonction prévue et ne remplit aucune fonction non prévue Garantir, de la manière la plus appropriée, que la configuration des données des systèmes PE répond aux exigences spécifiées pour la capabilité systématique du logiciel	modules logiciels	7.4.7	spécification d'essai des modules logiciels ; liste des codes source ; rapport de revue de code	résultats d'essai des modules logiciels ; modules logiciels vérifiés et soumis à l'essai
10.3	Conception et développement du logiciel	Essai d'intégration du logiciel: Vérifier que les exigences pour le logiciel de sécurité (en termes de fonctions de sécurité du logiciel requises et de capabilité systématique du logiciel) ont été remplies Montrer que tous les modules logiciels et composants/sous-systèmes logiciels interagissent correctement de façon à réaliser leur fonction prévue et ne réalisent aucune fonction non prévue Garantir, de la manière la plus appropriée, que la configuration des données des systèmes PE répond aux exigences spécifiées pour la capabilité systématique du logiciel	architecture du logiciel ; système logiciel	7.4.8	Spécification d'essai d'intégration du système logiciel	résultats d'essai d'intégration du système logiciel ; système logiciel vérifié et soumis à l'essai

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Tableau 1 *(suite)*

Phase du cycle de vie de sécurité		Objectifs	Domaine d'application	Paragraphe contenant les exigences	Entrées (information requise)	Sorties (information fournie)
Numéro de case de la Figure 4	Titre					
10.4	Intégration de l'électronique programmable (matériel et logiciel)	Intégrer le logiciel au matériel électronique programmable cible; Combiner le logiciel et le matériel de l'électronique programmable relative à la sécurité pour assurer leur compatibilité et satisfaire aux exigences du niveau d'intégrité de sécurité prévu	matériel électronique programmable; logiciel intégré	7.5.2	spécification d'essai d'intégration de l'architecture du logiciel; Spécification d'essai d'intégration du logiciel/système PE (également requise par la CEI 61508-2). électronique programmable intégrée	résultats d'essai d'intégration de l'architecture du logiciel; résultats de l'essai d'intégration de l'électronique programmable; électronique programmable intégrée, vérifiée et soumise à l'essai
10.5	Procédures d'exploitation et de modification du logiciel	Fournir des informations et des procédures concernant le logiciel nécessaires pour assurer que la sécurité fonctionnelle du système E/E/PE relatif à la sécurité est maintenue pendant l'exploitation et la modification	comme décrit ci-dessus	7.6.2	tous les éléments susmentionnés, si applicables	Procédures d'exploitation et de modification du logiciel
10.6	Validation de sécurité du logiciel	Assurer que le système intégré est conforme aux exigences spécifiées pour le logiciel de sécurité au niveau d'intégrité de sécurité prévu	comme décrit ci-dessus	7.7.2	Planification de la validation de sécurité du logiciel	résultats de la validation de sécurité du logiciel; logiciel validé
–	Modification du logiciel	Apporter des corrections, des améliorations ou des adaptations au logiciel validé en s'assurant du maintien de la capabilité systématique du logiciel requise	comme décrit ci-dessus	7.8.2	procédures de modification du logiciel; demande de modification du logiciel	résultats de l'analyse d'impact de la modification du logiciel; journal de modification du logiciel
–	Vérification du logiciel	Soumettre à l'essai et évaluer les sorties d'une phase donnée du cycle de vie de sécurité du logiciel pour assurer la conformité et la cohérence par rapport aux sorties et aux normes fournies en entrée de cette phase	dépend de la phase	7.9.2	plan de vérification approprié (dépend de la phase)	rapport de vérification approprié (dépend de la phase)
–	Evaluation de la sécurité fonctionnelle du logiciel	Evaluer et parvenir à un jugement concernant les aspects logiciels de la sécurité fonctionnelle atteinte par les systèmes E/E/PE relatifs à la sécurité	toutes les phases susmentionnées	8	plan d'évaluation de la sécurité fonctionnelle du logiciel	rapport d'évaluation de la sécurité fonctionnelle du logiciel

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.2 Spécification des exigences pour la sécurité du logiciel

NOTE Cette phase correspond à la case 10.1 de la Figure 4.

7.2.1 Objectifs

7.2.1.1 Le premier objectif des exigences du présent paragraphe est de spécifier les exigences pour le logiciel de sécurité comprenant les exigences relatives aux fonctions de sécurité du logiciel et les exigences relatives à la capabilité systématique du logiciel.

7.2.1.2 Le deuxième objectif des exigences du présent paragraphe est de spécifier les exigences relatives aux fonctions de sécurité du logiciel pour chaque système E/E/PE relatif à la sécurité nécessaire pour implémenter les fonctions de sécurité requises.

7.2.1.3 Le troisième objectif des exigences du présent paragraphe est de spécifier les exigences relatives à la capabilité systématique du logiciel pour chaque système E/E/PE relatif à la sécurité nécessaire pour atteindre le niveau d'intégrité de sécurité spécifié pour chaque fonction de sécurité attribuée à ce système E/E/PE relatif à la sécurité.

7.2.2 Exigences

NOTE 1 Ces exigences sont satisfaites, dans la plupart des cas, par la combinaison d'un logiciel générique embarqué et d'un logiciel spécifique à l'application. La combinaison des deux logiciels fournit les caractéristiques qui satisfont aux paragraphes suivants. La séparation exacte entre le logiciel générique et le logiciel spécifique à l'application dépend de l'architecture logicielle choisie (voir 7.4.3).

NOTE 2 Pour la sélection des techniques et mesures appropriées (voir Annexes A et B) visant à mettre en œuvre les exigences du présent article, il convient de tenir compte des propriétés suivantes (voir l'Annexe C pour les recommandations sur l'interprétation des propriétés, et l'Annexe F de la CEI 61508-7 pour les définitions informelles) concernant la spécification des exigences pour la sécurité du logiciel:

– complétude par rapport aux besoins de sécurité devant être traités par le logiciel;

– exactitude par rapport aux besoins de sécurité devant être traités par le logiciel;

– insensibilité aux anomalies de spécification intrinsèques, y compris exemption de toute ambiguïté;

– intelligibilité des exigences de sécurité;

– insensibilité aux perturbations préjudiciables des fonctions non liées à la sécurité avec les besoins de sécurité devant être traités par le logiciel;

– aptitude à fournir une base pour la vérification et la validation.

NOTE 3 Les besoins de sécurité devant être traités par le logiciel constituent l'ensemble des fonctions de sécurité et des exigences d'intégrité de sécurité correspondantes allouées aux fonctions logicielles par la conception du système E/E/PE. (L'ensemble complet des besoins de sécurité du système est un ensemble plus important qui comprend également les fonctions de sécurité qui ne dépendent pas du logiciel). La complétude de la spécification des exigences pour la sécurité du logiciel dépend de manière critique de l'efficacité des phases antérieures du cycle de vie du système.

7.2.2.1 Si les exigences pour le logiciel de sécurité ont déjà été spécifiées pour le système E/E/PE relatif à la sécurité (voir l'Article 7 de la CEI 61508-2), il n'est alors pas nécessaire de répéter la spécification des exigences pour la sécurité du logiciel.

7.2.2.2 La spécification des exigences pour le logiciel de sécurité doit découler des exigences de sécurité spécifiées du système E/E/PE relatif à la sécurité (voir Article 7 de la CEI 61508-2), et de toute exigence de la planification de sécurité (voir Article 6). Ces informations doivent être communiquées au développeur du logiciel.

NOTE 1 Cette exigence ne signifie pas une absence d'itération entre le développeur du système E/E/PE et le développeur du logiciel (CEI 61508-2 et CEI 61508-3). Au fur et à mesure que les exigences pour le logiciel de sécurité et que l'architecture logicielle se précisent, il est admis qu'il y ait des conséquences sur l'architecture matérielle du système E/E/PE, le maintien d'une étroite coopération entre les développeurs du matériel et du logiciel étant pour cette raison essentiel. Voir Figure 5.

NOTE 2 Lorsque la conception du logiciel intègre un logiciel préexistant réutilisable, ce dernier peut avoir été développé sans tenir compte de la spécification actuelle des exigences du système. Voir 7.4.2.12 pour les exigences relatives au logiciel préexistant permettant de satisfaire à la spécification des exigences pour la sécurité du logiciel.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.2.2.3 La spécification des exigences pour le logiciel de sécurité doit être suffisamment détaillée pour permettre d'obtenir l'intégrité de sécurité requise au cours de la conception et de l'implémentation (y compris toute exigence pour l'indépendance, voir 7.4.3 de la CEI 61508-2), et permettre une évaluation de la sécurité fonctionnelle.

NOTE Le niveau de détail de la spécification peut varier en fonction de la complexité de l'application. Une spécification appropriée de comportement fonctionnel peut comprendre des exigences relatives à la précision, à la synchronisation et aux performances, à la capacité, à la robustesse, à la tolérance à la surcharge, et autres propriétés de caractérisation de l'application spécifique.

7.2.2.4 Pour traiter l'indépendance, une analyse appropriée des défaillances de cause commune doit être réalisée. Lorsque des mécanismes de défaillance crédibles sont identifiés, des mesures défensives efficaces doivent être prises.

NOTE Voir l'Annexe F pour les techniques permettant de réaliser un aspect de l'indépendance du logiciel.

7.2.2.5 Le développeur du logiciel doit effectuer une revue des informations contenues en 7.2.2.2 afin de s'assurer que les exigences sont correctement spécifiées. Il doit tenir compte en particulier:

a) des fonctions de sécurité;

b) de la configuration ou de l'architecture du système;

c) des exigences d'intégrité de sécurité du matériel (électronique programmable, capteurs et actionneurs);

d) des exigences de capabilité systématique du logiciel;

e) des performances de capacité et de temps de réponse;

f) des interfaces équipement/opérateur, y compris un mauvais usage raisonnablement prévisible.

NOTE Il convient de tenir compte de la compatibilité avec toute application déjà existante.

7.2.2.6 S'ils ne sont pas déjà définis de manière appropriée dans les exigences de sécurité spécifiées du système E/E/PE relatif à la sécurité, tous les modes d'exploitation concernés de l'équipement commandé, du système E/E/PE et de tout équipement ou système relié au système E/E/PE doivent être détaillés dans les exigences spécifiées pour le logiciel de sécurité.

7.2.2.7 La spécification des exigences pour la sécurité du logiciel doit spécifier et documenter toute contrainte entre le matériel et le logiciel, relative à la sécurité ou importante pour celle-ci.

7.2.2.8 Dans les limites imposées par la conception de l'architecture du matériel E/E/PE, et compte tenu de l'éventuelle augmentation de complexité, la spécification des exigences pour la sécurité du logiciel doit tenir compte:

a) de l'auto-surveillance du logiciel (voir exemples dans la CEI 61508-7);

b) de la surveillance du matériel électronique programmable, des capteurs et des actionneurs;

c) de l'essai périodique des fonctions de sécurité pendant l'exploitation du système;

d) de la possibilité de soumettre à l'essai les fonctions de sécurité lorsque l'EUC est en exploitation;

e) des fonctions logicielles pour réaliser les essais périodiques et tous les essais de diagnostic afin de satisfaire à l'exigence d'intégrité de sécurité du système E/E/PE relatif à la sécurité.

NOTE L'augmentation de la complexité résultant des considérations ci-dessus peut nécessiter de réexaminer l'architecture.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.2.2.9 Lorsque le système E/E/PE relatif à la sécurité est requis pour exécuter des fonctions qui ne sont pas des fonctions de sécurité, les exigences spécifiées pour le logiciel de sécurité doivent alors clairement identifier ces fonctions.

NOTE Voir 7.4.2.8 et 7.4.2.9 pour les exigences relatives à la non-interférence entre les fonctions de sécurité et les fonctions qui ne sont pas des fonctions de sécurité.

7.2.2.10 La spécification des exigences pour la sécurité du logiciel doit exprimer les propriétés de sécurité requises du produit mais non du projet tel que traité par la planification de sécurité (voir l'Article 6 de la CEI 61508-1). En référence aux paragraphes 7.2.2.1 à 7.2.2.9, les points suivants doivent être spécifiés, selon le cas:

a) les exigences pour les fonctions de sécurité du logiciel suivantes:
   1) fonctions permettant à l'équipement commandé d'atteindre ou de maintenir un état de sécurité;
   2) fonctions liées à la détection, signalisation et gestion des anomalies du matériel électronique programmable;
   3) fonctions liées à la détection, signalisation et gestion des anomalies des capteurs et actionneurs;
   4) fonctions liées à la détection, signalisation et gestion des anomalies du logiciel proprement dit (auto-surveillance du logiciel);
   5) fonctions liées à l'essai périodique des fonctions de sécurité en ligne (c'est-à-dire dans l'environnement d'exploitation prévu);
   6) fonctions liées à l'essai périodique des fonctions de sécurité hors ligne (c'est-à-dire dans un environnement où l'utilisateur ne se fie pas à l'EUC pour sa fonction de sécurité);
   7) fonctions permettant la modification du système PE en toute sécurité;
   8) interfaces avec des fonctions non relatives à la sécurité;
   9) performances de capacité et de temps de réponse;
   10) interfaces entre le logiciel et le système PE;

       NOTE 1 Les interfaces comprennent à la fois les installations de programmation en ligne et hors ligne.

   11) communications relatives à la sécurité (voir 7.4.11 de la CEI 61508-2).
b) les exigences pour la capabilité systématique du logiciel:
   1) le ou les niveaux d'intégrité de sécurité pour chacune des fonctions mentionnées en a) ci-dessus;

      NOTE 2 Voir l'Annexe A de la CEI 61508-5 pour tout renseignement concernant l'allocation de l'intégrité de sécurité aux composants logiciels.

   2) les exigences d'indépendance entre les fonctions.

7.2.2.11 Lorsque les exigences pour la sécurité du logiciel sont exprimées ou mises en œuvre par les données de configuration, les données doivent être:

a) cohérentes avec les exigences de sécurité du système;
b) exprimées en termes de gamme admise et de combinaisons autorisées de ses paramètres d'exploitation;
c) définies de manière à être compatibles avec le logiciel de base (par exemple, séquence d'exécution, durée d'exécution, structures des données, etc.).

NOTE 1 Cette exigence relative aux données d'application s'applique tout particulièrement aux applications dirigées par les données. Elles sont caractérisées comme suit: le code source est préexistant et le principal objectif de l'activité de développement est de garantir que les données de configuration présentent le comportement requis par l'application. Il peut exister des dépendances complexes entre les éléments de données, et la validité des données peut varier dans le temps.

NOTE 2 Voir l'Annexe G pour des recommandations sur les systèmes dirigés par les données.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.2.2.12 Lorsque les données définissent l'interface entre le logiciel et les systèmes externes, les caractéristiques de performance suivantes doivent être prises en compte, en complément de celles définies en 7.4.11 de la CEI 61508-2:

a) la nécessité de cohérence en termes de définitions des données;

b) les valeurs invalides, hors gamme ou inopportunes;

c) le temps de réponse et le débit, y compris les conditions de chargement maximal;

d) la durée d'exécution dans le meilleur cas et le cas le plus défavorable, et l'interblocage;

e) le dépassement de capacité ou dépassement par valeurs inférieures de la capacité mémoire de données.

7.2.2.13 Les paramètres d'exploitation doivent être protégés contre:

a) les valeurs invalides, hors gamme ou inopportunes;

b) les modifications non autorisées;

c) l'altération.

NOTE 1 Il convient de considérer la protection contre les modifications non autorisées en tenant compte des mécanismes liés au logiciel et des mécanismes non liés au logiciel. A noter qu'une protection efficace contre les modifications non autorisées du logiciel peut avoir des effets préjudiciables sur la sécurité, par exemple, lorsque les modifications doivent être réalisées rapidement et dans des conditions de contrainte.

NOTE 2 Bien qu'une personne physique puisse faire partie d'un système relatif à la sécurité (voir l'Article 1 de la CEI 61508-1), les exigences concernant le facteur humain dans la conception de systèmes E/E/PE relatifs à la sécurité ne sont pas détaillées dans la présente norme. Il convient cependant de traiter, le cas échéant, les considérations d'ordre humain suivantes:

- Il convient qu'un système d'information de l'opérateur utilise la configuration graphique et la terminologie avec lesquelles les opérateurs sont familiarisés. Il convient qu'il soit clair, compréhensible et exempt de tous détails et/ou aspects inutiles;
- Il convient que les informations sur l'EUC affichées à l'opérateur correspondent étroitement à la disposition physique de l'EUC;
- Lorsqu'il est possible d'afficher plusieurs contenus à l'opérateur et/ou si les éventuelles actions de l'opérateur autorisent des interactions dont les conséquences ne peuvent être constatées à priori, il convient que les informations affichées comportent automatiquement, à chaque état d'un affichage ou d'une séquence d'action, des indications sur l'état auquel la séquence est parvenue, sur les opérations qu'il est possible de réaliser et sur les conséquences susceptibles d'être choisies.

7.3 Planification de la validation de sécurité du logiciel

NOTE 1 Cette phase correspond à la case 10.2 de la Figure 4.

NOTE 2 Le logiciel ne peut généralement pas être validé indépendamment de son matériel de base et environnement système.

7.3.1 Objectif

L'objectif des exigences du présent paragraphe est d'élaborer un plan pour valider la sécurité du logiciel.

7.3.2 Exigences

7.3.2.1 La planification doit être conduite afin de spécifier les étapes, qu'elles soient procédurales ou techniques, qui permettent de démontrer que le logiciel satisfait aux exigences de sécurité.

7.3.2.2 Le plan de validation de la sécurité du logiciel doit tenir compte des éléments suivants:

a) les détails concernant la date à laquelle la validation doit avoir lieu;

b) les détails concernant les personnes qui doivent effectuer la validation;

c) l'identification des modes d'exploitation concernés de l'EUC, comprenant:

--`,``,,`````````,,``,`,,``,`,`-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

1) les préparations d'utilisation, y compris la configuration et les réglages;
2) le démarrage, l'apprentissage, le mode automatique, le mode manuel, le mode semi-automatique, le régime établi,
3) la remise à zéro, l'arrêt et la maintenance;
4) les conditions anormales raisonnablement prévisibles et le mauvais usage de l'opérateur raisonnablement prévisible.

d) l'identification du logiciel de sécurité nécessitant une validation pour chaque mode d'exploitation de l'équipement commandé avant le début de la mise en service;

e) la stratégie technique de la validation (par exemple, les méthodes analytiques, les essais statistiques, etc.);

f) en accord avec le point e), les mesures (techniques) et les procédures qui doivent être utilisées pour confirmer que chaque fonction de sécurité est conforme aux exigences spécifiées pour les fonctions de sécurité du logiciel et aux exigences spécifiées pour la capabilité systématique du logiciel;

g) l'environnement requis dans lequel les activités de validation doivent se dérouler (par exemple, pour des essais, cet environnement peut comprendre les outils étalonnés ainsi que les équipements d'essai);

h) les critères d'acceptation ou de rejet;

i) les politiques et procédures d'évaluation des résultats de la validation, en particulier en ce qui concerne les défaillances.

NOTE Ces exigences sont basées sur les exigences générales décrites en 7.8 de la CEI 61508-1.

7.3.2.3 La validation doit fournir les raisons et la justification de la stratégie choisie. La stratégie technique concernant la validation d'un logiciel de sécurité doit comprendre les informations suivantes:

a) le choix entre des techniques manuelles ou automatiques, ou les deux;

b) le choix entre des techniques statiques ou dynamiques, ou les deux;

c) le choix entre des techniques analytiques ou statistiques, ou les deux.

d) le choix des critères d'acceptation basé sur des facteurs objectifs ou un jugement d'expert, ou les deux.

7.3.2.4 Dans le cadre de la procédure de validation d'un logiciel de sécurité, le domaine d'application et le contenu de la planification pour valider la sécurité du logiciel doivent être convenus avec l'évaluateur ou un représentant de l'évaluateur, si cela est requis par le niveau d'intégrité de sécurité (voir l'Article 8 de la CEI 61508-1). Cette procédure doit également contenir une clause concernant la présence de l'évaluateur durant l'essai.

7.3.2.5 Les critères d'acceptation ou de rejet de l'essai de validation du logiciel doivent comprendre:

a) les signaux d'entrée requis avec leurs séquences et leurs valeurs;

b) les signaux de sortie prévus avec leurs séquences et leurs valeurs; et

c) les autres critères d'acceptation, tels que l'utilisation de la mémoire, la synchronisation et les tolérances applicables aux valeurs.

7.4 Conception et développement du logiciel

NOTE Cette phase correspond à la case 10.3 de la Figure 4.

7.4.1 Objectifs

7.4.1.1 Le premier objectif des exigences du présent paragraphe est de créer une architecture de logiciel satisfaisant aux exigences spécifiées pour le logiciel de sécurité en fonction du niveau d'intégrité de sécurité requis.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.4.1.2 Le second objectif des exigences du présent paragraphe est d'évaluer les exigences imposées au logiciel par l'architecture matérielle du système E/E/PE relatif à la sécurité, y compris les conséquences des interactions matériel/logiciel de l'E/E/PE sur la sécurité de l'équipement commandé.

7.4.1.3 Le troisième objectif des exigences du présent paragraphe est de sélectionner un ensemble approprié d'outils d'aide à la vérification, la validation, l'évaluation et la modification, y compris les langages et les compilateurs, les interfaces système d'exécution, les interfaces utilisateurs et les formats et représentations des données pour le niveau d'intégrité de sécurité requis au cours du cycle de vie de sécurité complet du logiciel.

7.4.1.4 Le quatrième objectif des exigences du présent paragraphe est de concevoir et implémenter un logiciel qui répond aux exigences spécifiées pour le logiciel de sécurité en fonction du niveau d'intégrité de sécurité requis, qui soit analysable et vérifiable, et qui soit modifiable de façon sûre.

7.4.1.5 Le cinquième objectif des exigences du présent paragraphe est de vérifier que les exigences pour le logiciel de sécurité (en termes de fonctions de sécurité du logiciel requises et de capabilité systématique du logiciel) ont été remplies.

7.4.1.6 Le sixième objectif des exigences du présent paragraphe est de garantir, de la manière la plus appropriée, que la configuration des données des systèmes PE répond aux exigences spécifiées pour la capabilité systématique du logiciel.

7.4.2 Exigences générales

7.4.2.1 Suivant la nature du développement du logiciel, la responsabilité concernant le respect des exigences de conformité mentionnées en 7.4 peut incomber au fournisseur d'un environnement de programmation relatif à la sécurité (par exemple fournisseur d'automate programmable) seul, ou à l'utilisateur de cet environnement (par exemple, le développeur du logiciel d'application) seul, ou aux deux ensemble. L'attribution des responsabilités doit être déterminée pendant la planification de sécurité (voir Article 6).

NOTE Voir 7.4.3 pour les aspects de l'architecture des systèmes et des logiciels qui sont pertinents dans le choix de la répartition pratique des responsabilités.

7.4.2.2 Conformément au niveau d'intégrité de sécurité requis, la méthode de conception choisie doit inclure des dispositions qui facilitent:

a) l'abstraction, la modularité et autres caractéristiques pour maîtriser la complexité;
b) l'expression:
    1) de la fonctionnalité;
    2) du flux d'informations entre les composants;
    3) de la séquence et des informations temporelles;
    4) des contraintes de temps;
    5) des conflits et de la synchronisation d'accès aux ressources partagées;
    6) des structures de données et de leurs propriétés;
    7) des hypothèses de conception et de leurs liens;
    8) du traitement des exceptions;
    9) des hypothèses de conception (pré-conditions, post-conditions, invariants);
    10) des commentaires;
c) l'aptitude à représenter plusieurs vues de la conception, y compris les vues de structure et de comportement;
d) la compréhension par les développeurs et tous ceux qui ont besoin de comprendre la conception;

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

e) la vérification et la validation.

7.4.2.3 La testabilité et l'aptitude à la modification sûre doivent être prises en compte durant les activités de conception afin de faciliter la mise en œuvre de ces propriétés dans le système final relatif à la sécurité.

NOTE A titre d'exemple, on peut citer les modes de maintenance dans les industries de machines-outils et de procédé.

7.4.2.4 La méthode de conception choisie doit avoir des caractéristiques qui facilitent la modification du logiciel. Ces caractéristiques comprennent la modularité, le masquage d'informations et l'encapsulation.

NOTE Voir F.7.

7.4.2.5 Les représentations de la conception doivent être basées sur une notation clairement définie ou limitée à des caractéristiques clairement définies.

7.4.2.6 Dans la mesure du possible, la conception doit simplifier la partie du logiciel relative à la sécurité.

7.4.2.7 La conception du logiciel doit comprendre, en fonction du niveau d'intégrité de sécurité requis, une auto-surveillance des flux de contrôle et de données. En cas de détection d'une défaillance, les mesures appropriées doivent être prises.

7.4.2.8 Lorsque le logiciel doit implémenter à la fois des fonctions de sécurité et des fonctions qui ne sont pas de sécurité, le logiciel dans son ensemble doit alors être considéré comme relatif à la sécurité, à moins qu'il soit possible de démontrer au niveau de la conception que les défaillances des fonctions qui ne sont pas de sécurité n'affectent pas de manière préjudiciable les fonctions de sécurité.

7.4.2.9 Lorsque le logiciel doit implémenter des fonctions de sécurité correspondant à différents niveaux d'intégrité de sécurité, le logiciel dans son ensemble doit alors être considéré comme appartenant au niveau d'intégrité de sécurité le plus élevé, à moins qu'une indépendance suffisante des fonctions de sécurité correspondant aux différents niveaux d'intégrité de sécurité soit démontrée au niveau de la conception. Il doit être démontré que soit (1) l'indépendance est obtenue tant dans le domaine spatial que temporel, ou (2) que toute violation de l'indépendance est maîtrisée. La justification de l'indépendance des fonctions doit être documentée.

NOTE Voir l'Annexe F pour les techniques permettant de réaliser un aspect de l'indépendance du logiciel.

7.4.2.10 Lorsque la capabilité systématique d'un composant logiciel est inférieure au niveau d'intégrité de sécurité de la fonction de sécurité à laquelle le composant logiciel appartient, le composant doit être combiné avec d'autres composants de sorte que la capabilité systématique de la combinaison soit égale au niveau d'intégrité de sécurité de la fonction de sécurité.

7.4.2.11 Lorsqu'une fonction de sécurité est mise en œuvre au moyen d'une combinaison de composants logiciels de capabilité systématique connue, les exigences de capabilité systématique définies en 7.4.3 de la CEI 61508-2, doivent s'appliquer à la combinaison des composants.

NOTE Bien différencier (1) la *fonction de sécurité de bout en bout* prise en charge par un ou plusieurs composants de (2) la *fonction de sécurité du composant* de chaque composant de prise en charge. Lorsque deux composants sont combinés pour obtenir une capabilité systématique supérieure, il convient que chacun des composants appariés soit capable d'empêcher/pallier l'événement dangereux, mais il n'est pas nécessaire d'avoir des fonctions de sécurité du composant identiques. Il n'est également pas requis que chacun des composants appariés soit indépendamment capable d'assurer la fonctionnalité de sécurité totale sollicitée par la combinaison.

EXEMPLE Une commande de puissance de moteur électronique dont la *fonction de sécurité de bout en bout* consiste à "empêcher l'accélération non sollicitée". La *fonction de sécurité de bout en bout* est mise en œuvre par

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

deux processeurs. La *fonction de sécurité du composant* du contrôleur principal constitue le comportement idéal de sollicitation/réponse du régulateur. La *fonction de sécurité du composant* du processeur secondaire est de surveiller le fonctionnement d'un "ballon de sécurité" et d'appliquer un arrêt d'urgence si nécessaire. La combinaison des deux processeurs assure une plus grande confiance dans le fait que la fonction de sécurité de bout en bout "empêcher l'accélération non sollicitée" se réalise.

7.4.2.12 Lorsqu'un composant logiciel préexistant est réutilisé pour implémenter tout ou partie d'une fonction de sécurité, le composant doit satisfaire à la fois aux exigences a) et b) ci-dessous relatives à l'intégrité de sécurité systématique:

a) satisfaire aux exigences de l'un des parcours de conformité suivants:
   – Parcours $1_{S:}$ développement conforme. Conformité aux exigences de la présente norme pour l'évitement et la maîtrise des anomalies systématiques du logiciel;
   – Parcours $2_{S:}$ éprouvé par une utilisation antérieure. Fournir les preuves du caractère « éprouvé par une utilisation antérieure » du composant. Voir 7.4.10 de la CEI 61508-2;
   – Parcours 3_S: évaluation d'un développement non conforme. Conformité au 7.4.2.13.

NOTE 1 Les parcours 1_S, 2_S et 3_S représentent les parcours de conformité des composants définis en 7.4.2.2 c) de la CEI 61508-2, avec une référence particulière aux composants logiciels. Ils sont repris ici uniquement pour des raisons pratiques, et pour minimiser les références à la CEI 61508-2.

NOTE 2 Voir 3.2.8 de la CEI 61508-4. Le logiciel préexistant peut être un produit disponible dans le commerce ou avoir été développé par une organisation pour un produit ou système antérieur. Le logiciel préexistant peut avoir été ou non développé conformément aux exigences de la présente norme.

NOTE 3 Les exigences relatives aux composants préexistants s'appliquent à une bibliothèque d'exécution ou à un interpréteur.

b) fournir un manuel de sécurité (voir l'Annexe D de la CEI 61508-2 et l'Annexe D de la présente norme) qui spécifie une description suffisamment précise et complète du composant permettant de réaliser une évaluation de l'intégrité d'une fonction de sécurité spécifique qui dépend, en tout ou en partie, du composant logiciel préexistant.

NOTE 4 Le manuel de sécurité peut être déduit de la propre documentation du fournisseur du composant et des enregistrements du processus de développement du fournisseur du composant. Il peut également être créé ou complété par des activités de qualification supplémentaires entreprises par le développeur du système relatif à la sécurité ou par des tierces parties. Dans certains cas, il peut se révéler nécessaire de réaliser une ingénierie inverse pour élaborer une spécification ou une documentation de conception appropriée pour satisfaire aux exigences du présent article, soumises aux conditions légales en vigueur (par exemple, droits d'auteur ou droits de propriété intellectuelle).

NOTE 5 La justification du composant peut être réalisée pendant la planification de sécurité (voir Article 6).

7.4.2.13 Pour satisfaire au parcours $3_{S,}$ un composant logiciel préexistant doit satisfaire à toutes les exigences a) à i) suivantes:

a) La spécification des exigences pour la sécurité du logiciel pour le composant dans sa nouvelle application doit être documentée avec la même précision que celle requise par la présente norme pour tout composant relatif à la sécurité ayant la même capabilité systématique. La spécification des exigences pour la sécurité du logiciel doit couvrir le comportement fonctionnel et de sécurité applicable au composant dans sa nouvelle application et tel que spécifié en 7.2. Voir le Tableau A.1.

b) La justification de l'utilisation d'un composant logiciel doit fournir la preuve que les propriétés de sécurité souhaitées spécifiées dans les paragraphes référencés (c'est-à-dire 7.2.2, 7.4.3, 7.4.4, 7.4.5, 7.4.6, 7.4.7, 7.5.2, 7.7.2, 7.8.2, 7.9.2, et l'Article 8) ont été prises en compte, compte tenu des recommandations de l'Annexe C.

c) La conception du composant doit être documentée avec une précision suffisante pour démontrer la conformité à la spécification des exigences et à la capabilité systématique requise. Voir 7.4.3, 7.4.5 et 7.4.6, et les Tableaux A.2 et A.4 de l'Annexe A.

d) La preuve requise en 7.4.2.13 a) et 7.4.2.13 b) doit couvrir l'intégration du logiciel dans le matériel. Voir 7.5 et le Tableau A.6 de l'Annexe A.

e) Il doit être démontré que le composant a fait l'objet d'une vérification et d'une validation en adoptant une approche systématique avec essai et revue documentés de toutes les

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

parties de la conception et du code du composant. Voir 7.4.7, 7.4.8, 7.5, 7.7 et 7.9 et les Tableaux A.5 à A.7 et A.9 de l'Annexe A, ainsi que les tableaux associés de l'Annexe B.

NOTE 1 Une expérience positive d'exploitation peut être utilisée pour satisfaire aux exigences de boîte noire et essais probabilistes [voir Tableaux A.7 et B.3].

f) Lorsque le composant logiciel assure des fonctions qui ne sont pas utilisées ou ne sont pas nécessaires dans le système relatif à la sécurité, la preuve doit être donnée que les fonctions non utilisées et non nécessaires n'empêchent pas le système E/E/PE de satisfaire à ses exigences de sécurité.

NOTE 2 Les moyens de satisfaire à cette exigence comprennent:

- le retrait des fonctions de la construction;
- la désactivation des fonctions;
- une architecture système appropriée (par exemple, partitionnement, wrappers, diversité, vérification de la plausibilité des sorties);
- des essais étendus.

g) Il doit être démontré que tous les mécanismes de défaillance crédibles du composant logiciel ont été identifiés et que des mesures de prévention appropriées ont été mises en œuvre.

NOTE 3 Les mesures de prévention appropriées comprennent:

- une architecture système appropriée (par exemple, partitionnement, wrappers, diversité, plausibilité de vérification des sorties);
- le traitement des exceptions.

h) La planification de l'utilisation du composant doit identifier la configuration du composant logiciel, l'environnement d'exécution du logiciel et du matériel, et si nécessaire, la configuration du système de compilation / liaison.

i) La justification de l'utilisation du composant ne doit être valide que pour les applications qui respectent les hypothèses formulées dans le manuel de sécurité d'article conforme pour le composant (voir l'Annexe D de la CEI 61508-2 et l'Annexe D).

7.4.2.14 Le présent paragraphe 7.4.2 doit, dans la mesure du possible, s'appliquer aux données et aux langages de génération des données.

NOTE Voir l'Annexe G pour les recommandations sur les systèmes dirigés par les données.

a) Lorsqu'un système PE est constitué d'une fonctionnalité préexistante configurée par les données pour satisfaire aux exigences de l'application spécifique, la conception du logiciel d'application doit être adaptée au degré de configurabilité de l'application, à la fonctionnalité pré-fournie existante et à la complexité du système PE relatif à la sécurité.

b) Lorsque la fonctionnalité relative à la sécurité d'un système PE est essentiellement ou significativement déterminée par des données de configuration, des techniques et mesures appropriées doivent être utilisées pour empêcher l'introduction d'anomalies pendant la conception, la production, le chargement et la modification des données de configuration et pour assurer que les données de configuration correspondent à la logique de l'application.

c) La spécification des structures de données doit être:

1) cohérente avec les exigences fonctionnelles du système, y compris les données d'application;
2) exhaustive;
3) auto-cohérente;
4) formulée de sorte que les structures de données soient protégées contre toute altération ou corruption.

d) Lorsqu'un système PE est constitué d'une fonctionnalité préexistante configurée par les données pour satisfaire aux exigences de l'application spécifique, le processus de configuration proprement dit doit être documenté de manière appropriée.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.4.3 Exigences concernant la conception de l'architecture du logiciel

NOTE 1 L'architecture du logiciel définit les principaux composants et sous-systèmes du logiciel, leur méthode d'interconnexion, ainsi que la méthode de réalisation des attributs requis, en particulier l'intégrité de sécurité. Elle définit également le comportement global du logiciel ainsi que les méthodes d'interfaçage et d'interaction des composants logiciels. Les principaux composants du logiciel comprennent les systèmes d'exploitation, les bases de données, les sous-systèmes d'entrée/sortie de l'EUC, les sous-systèmes de communication, le ou les programmes d'application, les outils de programmation et de diagnostic, etc.

NOTE 2 Dans certains secteurs industriels, l'architecture du logiciel est appelée «description fonctionnelle» ou «spécification de conception fonctionnelle» (bien que ces documents puissent également couvrir le matériel).

NOTE 3 Dans certains contextes de programmation de l'application utilisateur, particulièrement dans les PLC (voir l'Annexe E de la CEI 61508-6), l'architecture du logiciel est donnée par le fournisseur comme une caractéristique standard du produit. Le fournisseur est tenu, dans le cadre de la présente norme, de garantir à l'utilisateur la conformité de ses produits aux exigences mentionnées en 7.4. L'utilisateur adapte l'automate programmable à l'application en utilisant les dispositifs de programmation standard, par exemple, le langage à contact. Les exigences de 7.4.3 à 7.4.8 demeurent applicables. L'exigence de définir et documenter l'architecture du logiciel peut être considérée comme une information utilisable par l'utilisateur pour sélectionner le PLC (ou équivalent) pour l'application.

NOTE 4 Du point de vue de la sécurité, la phase d'architecture du logiciel est la phase de développement de la stratégie de base concernant la sécurité pour le logiciel.

NOTE 5 Bien que la série CEI 61508 fixe des objectifs chiffrés de défaillance numériques pour les fonctions de sécurité exécutées par les systèmes E/E/PE relatifs à la sécurité, l'intégrité systématique n'est généralement pas quantifiée (voir 3.5.6 de la CEI 61508-4), et l'intégrité de sécurité du logiciel (voir 3.5.5 de la CEI 61508-4) est définie comme une capabilité systématique exprimée sur une échelle de confiance de 1 à 4 (voir 3.5.9 de la CEI 61508-4). La présente norme admet que le logiciel peut présenter une défaillance en sécurité ou non liée à la sécurité. L'architecture du système/logiciel doit être telle que des défaillances non liées à la sécurité d'un composant sont limitées par une contrainte d'architecture donnée, et il convient que les méthodes de développement tiennent compte de ces contraintes. La présente norme applique des techniques strictes de développement et de validation qui sont qualitativement cohérentes avec la capabilité systématique requise.

NOTE 6 Pour la sélection des techniques et mesures appropriées (voir Annexes A et B) visant à mettre en œuvre les exigences du présent article, il convient de tenir compte des propriétés suivantes (voir l'Annexe C pour les recommandations concernant l'interprétation des propriétés, et l'Annexe F de la CEI 61508-7 pour les définitions informelles) de la conception de l'architecture du logiciel:

– complétude par rapport à la spécification des exigences pour la sécurité du logiciel;
– exactitude par rapport à la spécification des exigences pour la sécurité du logiciel;
– insensibilité aux anomalies de conception intrinsèques;
– simplicité et intelligibilité;
– prévisibilité du comportement;
– conception vérifiable et pouvant être soumise à essai;
– tolérance aux anomalies;
– protection contre les défaillances de cause commune dues à des événements externes.

7.4.3.1 Suivant la nature du développement du logiciel, la responsabilité concernant le respect des exigences de conformité mentionnées en 7.4.4 peut incomber à plusieurs parties. La répartition de la responsabilité doit être documentée pendant la planification de sécurité (voir l'Article 6 de la CEI 61508-1).

7.4.3.2 La conception de l'architecture du logiciel doit être définie par le fournisseur et/ou le développeur du logiciel et une description détaillée de cette conception doit être fournie. La conception de l'architecture du logiciel doit:

a) sélectionner et justifier (voir 7.1.2.7) un ensemble intégré de techniques et de mesures nécessaires pendant les phases du cycle de vie de sécurité du logiciel pour satisfaire à la spécification des exigences pour le logiciel de sécurité en fonction du niveau d'intégrité de sécurité requis. Ces techniques et mesures comprennent les stratégies de conception du logiciel pour la tolérance aux anomalies (cohérente avec le matériel) et l'évitement des anomalies, incluant (lorsque cela est approprié) la redondance et la diversité;
b) être basée sur une partition en sous-systèmes/composants, avec les informations suivantes qui doivent être fournies pour chacun d'entre eux:
    1) le fait de déterminer si les sous-systèmes/composants ont été vérifiés précédemment ou non, et si oui, leurs conditions de la vérification;

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

2) le fait de déterminer si chaque sous-système/composant est relatif ou non à la sécurité;

3) la capacité systématique du logiciel du sous-système/composant.

c) déterminer toutes les interactions matériel/logiciel, évaluer et détailler leur importance;

NOTE Lorsque l'interaction logiciel/matériel est déjà déterminée par l'architecture du système, il suffit de faire référence à cette dernière.

d) utiliser pour représenter l'architecture une notation définie de façon non ambiguë ou limitée à des caractéristiques définies de façon non ambiguë;

e) sélectionner les caractéristiques de conception à utiliser pour maintenir l'intégrité de sécurité de toutes les données. Il est admis que ces données comprennent les données d'entrée/sortie de l'installation, les données de communication, les données d'interface opérateur, les données de maintenance et celles de la base de données interne;

f) spécifier les essais d'intégration de l'architecture du logiciel appropriés pour assurer que cette architecture satisfait à la spécification des exigences pour la sécurité du logiciel en fonction du niveau d'intégrité de sécurité requis.

7.4.3.3 Toute modification requise de la spécification des exigences de sécurité du système E/E/PE relatif à la sécurité (voir 7.2.2) après l'application du 7.4.3.2, doit recevoir l'accord du développeur du système E/E/PE et être documentée.

NOTE Il y a inévitablement itération entre l'architecture matérielle et l'architecture logicielle (voir Figure 5). Il est donc nécessaire de discuter avec le développeur du matériel des problèmes concernant, par exemple, la spécification d'essai pour l'intégration du logiciel et du matériel de l'électronique programmable (voir 7.5).

7.4.4 Exigences concernant les outils de support, y compris les langages de programmation

NOTE Pour la sélection des techniques et mesures appropriées (voir Annexes A et B) visant à mettre en œuvre les exigences du présent article, il convient de tenir compte des propriétés suivantes (voir l'Annexe C pour les recommandations sur l'interprétation des propriétés, et l'Annexe F de la CEI 61508-7 pour les définitions informelles) des outils de support:

– la mesure dans laquelle l'outil prend en charge la production du logiciel avec les propriétés de sécurité requises;

– la clarté du fonctionnement et de la fonctionnalité de l'outil;

– la précision et la répétabilité de la sortie.

7.4.4.1 Un outil de support direct du logiciel doit être considéré comme un composant logiciel du système relatif à la sécurité

NOTE Voir 3.2.10 et 3.2.11 de la CEI 61508-4 pour des exemples sur les outils directs et autonomes.

7.4.4.2 Les outils de support logiciel autonomes doivent être sélectionnés comme partie cohérente des activités de développement du logiciel.

NOTE 1 Voir 7.1.2 pour les exigences relatives au cycle de vie de développement du logiciel.

NOTE 2 Il convient d'utiliser des outils autonomes appropriés pour prendre en charge le développement du logiciel afin d'augmenter l'intégrité du logiciel en réduisant la probabilité d'introduction ou de non détection d'anomalies pendant le développement. Des exemples d'outils appropriés aux phases de cycle de vie de développement du logiciel comprennent:

a) des outils de transformation ou de traduction qui convertissent une représentation logicielle ou de conception (par exemple, du texte ou un diagramme) d'un niveau d'abstraction à un autre: outils d'affinement de conception, compilateurs, assembleurs, éditeurs de lien (linkers), binder (relieurs), chargeurs et outils de génération de code;

b) les outils de vérification et de validation tels que logiciels d'analyse de code statique, moniteurs de couverture des essais, systèmes d'aide de démonstration de théorèmes et simulateurs;

c) les outils de diagnostic utilisés pour maintenir et surveiller le logiciel dans les conditions d'exploitation;

d) les outils d'infrastructure tels que systèmes d'aide au développement;

e) les outils de gestion de la configuration tels que outils de gestion de version;

f) les outils de données d'application qui produisent ou maintiennent les données nécessaires pour définir les paramètres et instancier les fonctions du système. Ces données comprennent les paramètres de

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

fonction, les gammes d'instrument, les niveaux d'alarme et de déclenchement, les états de sortie à adopter en cas de défaillance, la configuration géographique.

NOTE 3 Il convient de sélectionner des outils de support autonomes destinés à être intégrés. Dans ce contexte, les outils sont intégrés s'ils fonctionnent ensemble de sorte que les sorties d'un outil présentent un contenu et un format appropriés à l'entrée automatique d'un autre outil, réduisant ainsi au minimum la possibilité d'introduire des erreurs humaines lors du remaniement des résultats intermédiaires.

NOTE 4 Il convient de sélectionner les outils de support autonomes en fonction de leur compatibilité avec les besoins de l'application, du système relatif à la sécurité et de la boîte à outils intégrée.

NOTE 5 Il convient de tenir compte de la disponibilité des outils appropriés pour alimenter les services qui sont nécessaires pendant toute la durée de vie du système E/E/PE relatif à la sécurité (par exemple, outils d'aide à la spécification, conception, implémentation, documentation et modification).

NOTE 6 Il convient de tenir compte de la compétence des utilisateurs des outils sélectionnés. Voir l'Article 6 de la CEI 61508-1 pour les exigences en matière de compétence.

7.4.4.3 La sélection des outils de support autonomes doit être justifiée.

7.4.4.4 Tous les outils de support autonomes des classes T2 et T3 et doivent disposer d'une spécification ou d'un manuel produit définissant clairement le comportement de l'outil et fournissant toutes instructions ou contraintes concernant son utilisation. Voir 7.1.2 pour les exigences relatives au cycle de vie du développement du logiciel, et 3.2.11 de la CEI 61508-4 pour les catégories d'outils de support autonomes de logiciel.

NOTE Cette « spécification » ou ce « manuel produit » ne constitue pas un manuel de sécurité d'article conforme (voir l'Annexe D de la CEI 61508-2 ainsi que de la présente norme) pour l'outil proprement dit. Le concept de manuel de sécurité d'article conforme ne concerne qu'un composant préexistant incorporé au système relatif à la sécurité exécutable. Lorsqu'un composant préexistant a été généré par un outil T3, puis incorporé au système relatif à la sécurité exécutable, il convient alors d'inclure toute information pertinente issue de la « spécification ou manuel produit » de l'outil dans le manuel de sécurité d'article conforme permettant ainsi de réaliser une évaluation de l'intégrité d'une fonction de sécurité spécifique qui dépend, en tout ou en partie, du composant incorporé.

7.4.4.5 Pour les outils de support autonomes de classes T2 et T3, une évaluation doit être réalisée pour déterminer le niveau de confiance accordée aux outils, et les mécanismes de défaillance potentielle des outils susceptibles d'affecter le logiciel exécutable. Lorsque les mécanismes de défaillance de ce type sont identifiés, des mesures de prévention appropriées doivent être prises.

NOTE 1 Une analyse HAZOP de logiciel représente une technique d'analyse des conséquences de défaillances potentielles d'outil logiciel.

NOTE 2 Les exemples de mesures de prévention comprennent: évitement de bogues connus, utilisation limitée de la fonctionnalité de l'outil, vérification de la sortie de l'outil, utilisation de divers outils pour les mêmes fins.

7.4.4.6 Pour chaque outil de classe T3, il doit être démontré que l'outil est conforme à sa spécification ou à son manuel. La preuve peut être fondée sur une combinaison appropriée de l'historique de l'utilisation satisfaisante dans des environnements similaires et pour des applications similaires (dans le cadre de l'organisation ou d'autres organisations), et de la validation de l'outil tel que spécifié en 7.4.4.7.

NOTE 1 Un historique de version peut fournir l'assurance de la maturité de l'outil et un enregistrement des erreurs / ambiguïtés qu'il convient de prendre en compte lorsque l'outil est utilisé dans le nouvel environnement de développement.

NOTE 2 La preuve répertoriée pour l'outil T3 peut également être utilisée pour les outils T2 pour ce qui concerne le jugement porté sur la précision de leurs résultats.

7.4.4.7 Les résultats de la validation de l'outil doivent être documentés et couvrir les résultats suivants:

a) un enregistrement chronologique des activités de validation;

b) la version du manuel produit de l'outil utilisée;

c) les fonctions de l'outil validées;

d) les outils et l'équipement utilisés;

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

e) les résultats de l'activité de validation; les résultats documentés de la validation doivent spécifier que le logiciel a satisfait à la validation ou les raisons de son échec;

f) les cas d'essai et leurs résultats pour une analyse ultérieure;

g) les divergences entre les résultats prévus et les résultats réels.

7.4.4.8 En l'absence de preuve de conformité au 7.4.4.6, des mesures efficaces doivent être prises pour maîtriser les défaillances du système relatif à la sécurité exécutable résultant d'anomalies imputables à l'outil.

NOTE Un exemple de mesure serait la génération de divers codes redondants qui permettent la détection et la maîtrise des défaillances du système relatif à la sécurité exécutable, suite à des anomalies introduites par un traducteur dans ce même système.

7.4.4.9 La compatibilité des outils d'une boîte à outils intégrée doit être vérifiée.

7.4.4.10 Dans les limites prescrites par le niveau d'intégrité de sécurité, la représentation logicielle ou de conception (y compris un langage de programmation) sélectionnée doit:

a) comprendre un traducteur évalué en vue d'établir son aptitude à l'utilisation prévue, y compris, le cas échéant, une évaluation par rapport à des étalons internationaux ou nationaux;

b) utiliser uniquement les caractéristiques de langage définies;

c) correspondre aux caractéristiques de l'application;

d) comporter des caractéristiques qui facilitent la détection des erreurs de conception ou de programmation;

e) supporter des caractéristiques adaptées à la méthode de conception.

NOTE 1 Un langage de programmation constitue une classe de représentations logicielles ou de conception. Un traducteur convertit une représentation logicielle ou de conception (par exemple, du texte ou un diagramme) d'un niveau d'abstraction à un autre. Les exemples de traducteurs comprennent: des outils d'affinement de conception, compilateurs, assembleurs, éditeurs de lien (linkers), binders (relieurs), chargeurs et outils de génération de code.

NOTE 2 Un traducteur peut être évalué pour un projet d'application spécifique ou pour une classe d'applications. Dans le dernier cas, il convient de mettre à disposition de l'utilisateur de l'outil toutes les informations nécessaires sur l'outil (la "spécification ou manuel produit", voir 7.4.4.4) concernant son utilisation prévue et appropriée. L'évaluation de l'outil pour un projet spécifique peut ensuite être réduite pour vérifier l'aptitude générale de l'outil pour le projet et la conformité à la "spécification ou manuel produit" (c'est-à-dire la bonne utilisation de l'outil). Une bonne utilisation peut comprendre des activités de vérification supplémentaires dans le cadre du projet spécifique.

NOTE 3 Une série de validations (c'est-à-dire un ensemble de programmes d'essai dont la traduction correcte est connue par anticipation) peut être utilisée pour évaluer l'aptitude à l'emploi d'un traducteur selon des critères définis. Il convient que ces critères comprennent des exigences fonctionnelles et non fonctionnelles. Pour ce qui concerne les exigences fonctionnelles du traducteur, des essais dynamiques peuvent constituer la principale technique de validation. Il convient de réaliser, si possible, une série d'essais automatiques.

7.4.4.11 Lorsque les spécifications de 7.4.4.10 ne peuvent pas être entièrement satisfaites, l'aptitude à l'emploi du langage et toutes mesures supplémentaires concernant des lacunes identifiées du langage doivent être justifiées.

7.4.4.12 Les langages de programmation pour le développement de tous les logiciels relatifs à la sécurité doivent être utilisés selon une règle de codage de langage de programmation appropriée.

NOTE Voir la CEI 61508-7 pour les recommandations sur les règles de codage liées à la sécurité du logiciel.

7.4.4.13 Les règles de codage d'un langage de programmation doivent spécifier une bonne pratique de la programmation, interdire les caractéristiques de langage peu sûres (par exemple, les caractéristiques du langage non définies, les conceptions non structurées, etc.), promouvoir l'intelligibilité des codes, faciliter la vérification et les essais, et spécifier les procédures pour documenter le code source. Dans toute la mesure du possible, le code source doit contenir les informations suivantes:

a) l'entité légale (par exemple, la société, le ou les auteurs, etc.);

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

b) une description;

c) les entrées et sorties;

d) l'historique de la gestion de configuration.

7.4.4.14 Dans le cas d'une génération de code automatique ou d'une traduction automatique similaire, la pertinence du traducteur automatique pour le développement du système relatif à la sécurité doit être évaluée dans le cadre du cycle de vie de développement au point où les outils d'aide au développement sont sélectionnés.

7.4.4.15 Lorsque les outils de support autonomes des classes T2 et T3 génèrent des articles dans le référentiel de configuration, la gestion de configuration doit assurer que les informations relatives aux outils sont enregistrées dans le référentiel de configuration. Ceci comprend en particulier:

a) l'identification de l'outil et sa version;

b) l'identification des articles de référentiel de configuration pour lesquels la version de l'outil a été utilisée;

c) le mode d'utilisation de l'outil (y compris les paramètres d'outil, les options et les scripts sélectionnés) pour chaque article de référentiel de configuration.

NOTE Le présent article a pour objectif de permettre la reconstruction du référentiel.

7.4.4.16 La gestion de configuration doit assurer que pour les outils des classes T2 et T3, seules les versions justifiées sont utilisées.

7.4.4.17 La gestion de configuration doit assurer que seuls des outils compatibles entre eux et avec le système relatif à la sécurité sont utilisés.

NOTE Le matériel du système relatif à la sécurité peut également imposer des contraintes de compatibilité aux outils logiciels, par exemple, un émulateur de processeur doit représenter un modèle précis de l'électronique réelle du processeur.

7.4.4.18 Chaque nouvelle version d'outil de support autonome doit être justifiée. Cette justification peut être fondée sur une preuve fournie pour une version antérieure sous réserve qu'elle soit suffisante pour ce qui concerne les éléments suivants:

a) les différences fonctionnelles (si elles existent) n'affectent pas la compatibilité de l'outil avec le reste de la boîte à outils; et

b) la nouvelle version n'est vraisemblablement pas susceptible de comporter de nouvelles anomalies inconnues significatives.

NOTE La preuve selon laquelle la nouvelle version n'est vraisemblablement pas susceptible de comporter de nouvelles anomalies inconnues significatives peut être fondée sur (1) une identification crédible des modifications apportées, (2) une analyse des actions de vérification et de validation réalisées sur la nouvelle version, et (3) toute expérience d'exploitation existante d'autres utilisateurs et pertinente pour la nouvelle version.

7.4.4.19 Suivant la nature du développement du logiciel, la responsabilité concernant le respect des exigences de conformité mentionnées en 7.4.4 peut incomber à plusieurs parties. La répartition de la responsabilité doit être documentée pendant la planification de sécurité (voir l'Article 6 de la CEI 61508-1).

7.4.5 Exigences concernant la conception détaillée et le développement – conception du système logiciel

NOTE 1 La conception détaillée est définie ici comme étant la conception du système logiciel: la partition des principaux composants de l'architecture en un système de modules logiciels, la conception des modules logiciels individuels et le codage. Dans les petites applications, la conception du système logiciel et celle de l'architecture peuvent être combinées.

NOTE 2 La nature de la conception détaillée et du développement varie selon la nature des activités de développement du logiciel et de l'architecture logicielle (voir 7.4.3). Dans certains contextes de programmation des applications, par exemple, langage à contacts et blocs fonctionnels, la conception détaillée peut être considérée comme de la configuration plutôt que de la programmation. Cependant, concevoir le logiciel de façon structurée

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

constitue toujours une bonne pratique, incluant l'organisation du logiciel en une structure modulaire séparant (autant que possible) les parties relatives à la sécurité; une vérification des limites et d'autres caractéristiques fournissant une protection contre les erreurs d'entrée de données; l'utilisation de modules logiciels déjà vérifiés et la fourniture d'une conception facilitant les modifications ultérieures du logiciel.

NOTE 3 Pour la sélection des techniques et mesures appropriées (voir Annexes A et B) visant à mettre en œuvre les exigences du présent article, il convient de tenir compte des propriétés suivantes (voir l'Annexe C pour les recommandations sur l'interprétation des propriétés, et l'Annexe F de la CEI 61508-7 pour les définitions informelles) de la conception et du développement:

– complétude par rapport à la spécification des exigences pour la sécurité du logiciel;
– exactitude par rapport à la spécification des exigences pour la sécurité du logiciel;
– insensibilité aux anomalies de conception intrinsèques;
– simplicité et intelligibilité;
– prévisibilité du comportement;
– conception vérifiable et pouvant être soumise à essai;
– tolérance aux anomalies / détection d'anomalies;
– insensibilité aux défaillances de cause commune.

7.4.5.1 Suivant la nature du développement du logiciel, la responsabilité concernant le respect des exigences de conformité mentionnées en 7.4.5 peut incomber à plusieurs parties. La répartition de la responsabilité doit être documentée pendant la planification de sécurité (voir l'Article 6 de la CEI 61508-1).

7.4.5.2 Les informations suivantes doivent être disponibles avant le début du processus de conception détaillée: la spécification des exigences relatives au système E/E/PE relatif à la sécurité; la conception de l'architecture logicielle; le plan de validation de la sécurité du logiciel.

7.4.5.3 Le logiciel doit être produit en vue d'obtenir la modularité, la testabilité et la capacité de modification sûre.

7.4.5.4 Pour chaque composant/sous-système principal identifié dans la conception de l'architecture logicielle, un affinement supplémentaire de la conception doit être basé sur la partition en modules logiciels (c'est-à-dire la spécification de conception du système logiciel). La conception de chaque module logiciel et la vérification à appliquer à chaque module doivent être spécifiées.

NOTE 1 Pour les composants logiciels préexistants, voir 7.4.2.

NOTE 2 La vérification comprend les essais et l'analyse.

7.4.5.5 Les essais d'intégration du système logiciel appropriés doivent être spécifiés en vue d'assurer que le système logiciel satisfait à la spécification des exigences pour la sécurité du logiciel, au niveau d'intégrité de sécurité requis.

7.4.6 Exigences concernant le codage

NOTE Selon l'étendue requise par le niveau d'intégrité de sécurité, le code source doit posséder les propriétés suivantes (voir les Annexes A et B pour les techniques spécifiques, et voir l'Annexe C pour les recommandations sur l'interprétation des propriétés) qu'il convient de prendre en considération:

- être lisible, compréhensible et pouvant être soumis à essai;
- satisfaire aux exigences spécifiées pour la conception des modules logiciels (voir 7.4.5);
- satisfaire aux exigences spécifiées dans les règles de codage (voir 7.4.4);
- satisfaire à toutes les exigences pertinentes identifiées pendant la planification de sécurité (voir l'Article 6).

7.4.6.1 Chaque module de code logiciel doit faire l'objet d'une revue. Lorsque le code est produit par un outil automatique, les exigences spécifiées en 7.4.4 doivent être satisfaites. Lorsque le code source est constitué d'un logiciel préexistant réutilisé, les exigences spécifiées en 7.4.2 doivent être satisfaites.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

NOTE La revue de code constitue une activité de vérification (voir 7.9). Elle peut prendre la forme d'une inspection du code: (1) par une personne physique; (2) par une lecture croisée du logiciel (voir CEI 61508-7 C.5.15); ou (3) par une inspection formelle (voir CEI 61508-7 C.5.14), par ordre croissant de sévérité.

7.4.7 Exigences concernant l'essai des modules logiciels

NOTE 1 Vérifier par essai que le module logiciel satisfait pleinement à la spécification d'essai correspondante constitue une activité de vérification (voir 7.9). La combinaison de la revue de code et de l'essai de module logiciel assure qu'un module logiciel satisfait à la spécification associée, c'est-à-dire qu'il est vérifié.

NOTE 2 Pour la sélection des techniques et mesures appropriées (voir Annexes A et B) visant à mettre en œuvre les exigences du présent article, il convient de tenir compte des propriétés suivantes (voir l'Annexe C pour les recommandations sur l'interprétation des propriétés, et l'Annexe F de la CEI 61508-7 pour les définitions informelles) de l'essai des modules logiciels:

– complétude de l'essai par rapport à la spécification de conception du logiciel;

– exactitude de l'essai par rapport à la spécification de conception du logiciel (réalisation satisfaisante);

– répétabilité;

– configuration d'essai définie avec précision.

7.4.7.1 Chaque module logiciel doit être vérifié comme requis par la spécification d'essai du module logiciel élaborée pendant la conception du système logiciel (voir 7.4.5).

NOTE La vérification comprend les essais et l'analyse.

7.4.7.2 Cette vérification doit montrer si chaque module logiciel remplit ou non la fonction prévue et ne remplit aucune fonction non prévue.

NOTE 1 Cela n'implique pas l'essai de toutes les combinaisons d'entrées, ni celui de toutes les combinaisons de sorties. L'essai de toutes les classes d'équivalence ou des essais basés sur la structure peuvent se révéler suffisants. Il est admis que l'analyse des valeurs aux limites ou l'analyse de flux de contrôle réduisent les cas d'essai à un nombre acceptable. Les programmes analysables rendent les exigences plus faciles à satisfaire. Pour ces techniques, voir l'Annexe C de la CEI 61508-7.

NOTE 2 L'étendue de ces essais peut être réduite en cas de développement à l'aide de méthodes formelles, de preuves formelles ou d'assertions. Pour ces techniques, voir l'Annexe C de la CEI 61508-7.

NOTE 3 Bien que l'intégrité systématique soit généralement non quantifiée (voir 3.5.6 de la CEI 61508-4), une preuve statistique quantifiée (par exemple, essai statistique, croissance de la fiabilité) est acceptable si toutes les conditions pertinentes applicables à la preuve valide du point de vue statistique sont remplies, par exemple, voir l'Annexe D de la CEI 61508-7.

NOTE 4 Si le module est suffisamment simple pour pouvoir réaliser un essai complet, ceci peut alors constituer le moyen le plus efficace pour démontrer la conformité.

7.4.7.3 Les résultats de l'essai des modules logiciels doivent être documentés.

7.4.7.4 Les procédures pour les actions correctives suite à l'échec de l'essai doivent être spécifiées.

7.4.8 Exigences concernant l'essai d'intégration du logiciel

NOTE Vérifier par essai que le logiciel est correctement intégré constitue une activité de vérification (voir 7.9).

7.4.8.1 Les essais d'intégration du logiciel doivent être spécifiés durant la phase de conception et de développement (voir 7.4.5).

7.4.8.2 La spécification d'essai d'intégration du système logiciel doit comprendre:

a) la répartition du logiciel en ensembles d'intégration gérables;

b) les cas d'essai et données d'essai;

c) les types d'essais à effectuer;

d) l'environnement, les outils, la configuration et les programmes d'essai;

e) les critères suivant lesquels la réalisation de l'essai est jugée;

f) les procédures pour les actions correctives en cas d'échec de l'essai.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.4.8.3 Le logiciel doit être soumis à l'essai conformément aux essais d'intégration de logiciel mentionnés dans la spécification d'essai d'intégration du système logiciel. Ces essais doivent montrer que tous les modules logiciels et composants/sous-systèmes logiciels interagissent correctement de façon à réaliser leur fonction prévue et ne réalisent aucune fonction non prévue.

NOTE 1 Cela n'implique pas l'essai de toutes les combinaisons d'entrées, ni celui de toutes les combinaisons de sorties. L'essai de toutes les classes d'équivalence ou des essais basés sur la structure peuvent se révéler suffisants. Il est admis que l'analyse des valeurs aux limites ou l'analyse de flux de contrôle réduisent les cas d'essai à un nombre acceptable. Les programmes analysables rendent les exigences plus faciles à satisfaire. Pour ces techniques, voir l'Annexe C de la CEI 61508-7.

NOTE 2 L'étendue de ces essais peut être réduite en cas de développement à l'aide de méthodes formelles, de preuves formelles ou d'assertions. Pour ces techniques, voir l'Annexe C de la CEI 61508-7.

NOTE 3 Bien que l'intégrité systématique soit généralement non quantifiée (voir 3.5.6 de la CEI 61508-4,), une preuve statistique quantifiée (par exemple, essai statistique, croissance de la fiabilité) est acceptable si toutes les conditions pertinentes applicables à la preuve valide du point de vue statistique sont remplies, par exemple, voir l'Annexe D de la CEI 61508-7.

7.4.8.4 Les résultats de l'essai d'intégration du logiciel doivent être documentés en établissant les résultats de l'essai, et en indiquant si les objectifs et les critères d'essai ont été atteints. En cas d'échec de l'essai d'intégration, les raisons de l'échec doivent être documentées.

7.4.8.5 Lors de l'intégration du logiciel, toute modification de ce dernier doit faire l'objet d'une analyse d'impact qui doit déterminer tous les modules logiciels touchés et les activités nécessaires de reconception et de revérification.

7.5 Intégration de l'électronique programmable (matériel et logiciel)

NOTE Cette phase correspond à la case 10.4 de la Figure 4.

7.5.1 Objectifs

7.5.1.1 Le premier objectif des exigences du présent paragraphe est d'intégrer le logiciel au matériel de l'électronique programmable cible.

7.5.1.2 Le deuxième objectif des exigences du présent paragraphe est de combiner le logiciel et le matériel de l'électronique programmable relative à la sécurité pour assurer leur compatibilité et pour satisfaire aux exigences du niveau d'intégrité de sécurité prévu.

NOTE 1 Vérifier que le logiciel est correctement intégré au matériel de l'électronique programmable constitue une activité de vérification (voir 7.9).

NOTE 2 Selon la nature de l'application, ces activités peuvent être combinées avec celles mentionnées en 7.4.8.

7.5.2 Exigences

NOTE Pour la sélection des techniques et mesures appropriées (voir Annexes A et B) visant à mettre en œuvre les exigences du présent article, il convient de tenir compte des propriétés suivantes (voir l'Annexe C pour les recommandations sur l'interprétation des propriétés, et l'Annexe F de la CEI 61508-7 pour les définitions informelles) de l'intégration:

– complétude de l'intégration par rapport aux spécifications de conception;

– exactitude de l'intégration par rapport aux spécifications de conception (réalisation satisfaisante);

– répétabilité;

– configuration d'intégration définie avec précision.

7.5.2.1 Les essais d'intégration doivent être spécifiés pendant la phase de conception et de développement (voir 7.4.3) afin d'assurer la compatibilité du matériel et du logiciel de l'électronique programmable relative à la sécurité.

NOTE Une coopération étroite avec le développeur du système E/E/PE peut être requise pour le développement des essais d'intégration.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.5.2.2 La spécification des essais d'intégration du logiciel/électronique programmable (matériel et logiciel) doit indiquer les points suivants:

a) la division du système en niveaux d'intégration;

b) les cas d'essai et données d'essai;

c) les types d'essais à effectuer;

d) la description de l'environnement d'essai comprenant les outils, le logiciel support et la configuration;

e) les critères suivant lesquels la réalisation de l'essai est jugée.

7.5.2.3 La spécification des essais d'intégration du logiciel/électronique programmable (matériel et logiciel) doit faire la distinction entre les activités que le développeur est capable d'effectuer sur son propre site et les activités nécessitant l'accès au site utilisateur.

7.5.2.4 La spécification des essais d'intégration du logiciel/électronique programmable (matériel et logiciel) doit faire la distinction entre les activités suivantes:

a) l'intégration du système logiciel au matériel électronique programmable cible;

b) l'intégration E/E/PE, c'est-à-dire l'ajout d'interfaces telles que des capteurs et des actionneurs;

c) l'application du système E/E/PE relatif à la sécurité à l'EUC.

NOTE Les points b) et c) sont traités dans la CEI 61508-1 et la CEI 61508-2 et sont cités ici afin de placer l'élément a) dans son contexte et fournir l'intégralité des informations. Ils ne relèvent généralement pas de la responsabilité des développeurs de logiciel.

7.5.2.5 Le logiciel doit être intégré au matériel de l'électronique programmable relative à la sécurité conformément à la spécification des essais d'intégration du logiciel/électronique programmable (matériel et logiciel).

7.5.2.6 Lors de l'essai d'intégration de l'électronique programmable relative à la sécurité (matériel et logiciel), toute modification apportée au système intégré doit être soumise à une analyse d'impact. Cette analyse doit déterminer tous les modules logiciels touchés et les activités de revérification nécessaires.

7.5.2.7 Les cas d'essai et les résultats prévus doivent être documentés pour analyse ultérieure.

7.5.2.8 L'essai d'intégration de l'électronique programmable relative à la sécurité (matériel et logiciel) doit être documenté, en établissant les résultats de l'essai et en indiquant si les objectifs et les critères d'essai ont été atteints. En cas d'échec, les raisons doivent être documentées. Toute modification ou tout changement résultant pour le logiciel doit être soumis à une analyse d'impact qui doit identifier tous les composants/modules logiciels touchés et les activités de reconception et revérification nécessaires.

7.6 Procédures d'exploitation et de modification du logiciel

NOTE Cette phase correspond à la case 10.5 de la Figure 4.

7.6.1 Objectif

L'objectif des exigences du présent paragraphe est de fournir des informations et procédures concernant le logiciel nécessaire pour assurer que la sécurité fonctionnelle du système E/E/PE relatif à la sécurité est maintenue pendant l'exploitation et la modification.

7.6.2 Exigences

Les exigences sont indiquées en 7.6 de la CEI 61508-2 et en 7.8 de la présente norme.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

NOTE Dans la présente norme, le logiciel (à la différence du matériel) ne peut être maintenu: il est toujours modifié.

7.7 Validation de sécurité du logiciel

NOTE 1 Cette phase correspond à la case 10.6 de la Figure 4.

NOTE 2 Le logiciel ne peut généralement pas être validé indépendamment de son logiciel de base et de l'environnement système.

7.7.1 Objectif

L'objectif des exigences du présent paragraphe est d'assurer que le système intégré est conforme aux exigences spécifiées pour la sécurité du logiciel au niveau d'intégrité de sécurité requis.

7.7.2 Exigences

NOTE Pour la sélection des techniques et mesures appropriées (voir Annexes A et B) visant à mettre en œuvre les exigences du présent article, il convient de tenir compte des propriétés suivantes (voir l'Annexe C pour les recommandations sur l'interprétation des propriétés, et l'Annexe F de la CEI 61508-7 pour les définitions informelles) de la validation de sécurité:

– complétude de la validation par rapport à la spécification de conception du logiciel;

– exactitude de la validation par rapport à la spécification de conception du logiciel (réalisation satisfaisante);

– répétabilité;

– configuration de validation définie avec précision.

7.7.2.1 Si la conformité aux exigences pour le logiciel de sécurité a déjà été établie dans la planification de la validation de sécurité du système E/E/PE relatif à la sécurité (voir 7.7 de la CEI 61508-2), il n'est alors pas nécessaire de répéter la validation.

7.7.2.2 Les activités de validation doivent être effectuées comme il est spécifié dans le plan de validation de sécurité du logiciel.

7.7.2.3 Suivant la nature du développement du logiciel, la responsabilité concernant le respect des exigences de conformité mentionnées en 7.7 peut incomber à plusieurs parties. La répartition de la responsabilité doit être documentée pendant la planification de sécurité (voir l'Article 6 de la CEI 61508-1).

7.7.2.4 Les résultats de la validation de sécurité du logiciel doivent être documentés.

7.7.2.5 Pour chaque fonction de sécurité, la validation de sécurité du logiciel doit documenter les résultats suivants:

a) un enregistrement chronologique des activités de validation permettant de retracer la séquence des activités;

   NOTE S'agissant de l'enregistrement des résultats d'essai, il est important de pouvoir retracer la séquence des activités. Cette exigence s'applique essentiellement au fait de retracer une séquence d'activités et non à produire une liste de documents temporisés/datés.

b) la version du plan de validation de sécurité du logiciel utilisée (voir 7.3);

c) la fonction de sécurité soumise à validation (par essai ou analyse), avec la référence au plan de validation de sécurité du logiciel;

d) les outils et les équipements utilisés avec les données d'étalonnage;

e) les résultats de l'activité de validation;

f) les divergences entre les résultats prévus et les résultats réels.

7.7.2.6 Lorsque les résultats réels ne correspondent pas aux résultats prévus, l'analyse effectuée ainsi que les décisions de poursuivre la validation ou d'émettre une demande de modification et de revenir à une phase antérieure du cycle de vie du développement, doivent être documentées en tant que résultats de la validation de sécurité du logiciel.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

NOTE Les exigences mentionnées de 7.7.2.2 à 7.7.2.6 sont fondées sur les exigences générales mentionnées en 7.14 de la CEI 61508-1.

7.7.2.7 La validation de sécurité du logiciel doit satisfaire aux exigences suivantes:

a) l'essai doit constituer la méthode principale de validation pour le logiciel; il est permis de compléter les activités de validation par des opérations d'analyse, d'animation et de modélisation;

b) le logiciel doit être exécuté en simulant:

   1) les signaux d'entrée présents en exploitation normale;

   2) les événements prévus;

   3) les conditions non désirées nécessitant une action du système;

c) le fournisseur et/ou le développeur (ou les différentes parties responsables de la conformité) doit fournir au développeur du système l'accessibilité aux résultats documentés de la validation de sécurité du logiciel ainsi que toute documentation utile afin de permettre à son produit de satisfaire aux exigences de la CEI 61508-1 et de la CEI 61508-2.

7.7.2.8 Les outils logiciels doivent satisfaire aux exigences de 7.4.4.

7.7.2.9 Les résultats de la validation de sécurité du logiciel doivent satisfaire aux exigences suivantes:

a) les essais doivent montrer que toutes les exigences spécifiées pour le logiciel de sécurité (voir 7.2) sont correctement satisfaites et que le logiciel ne remplit aucune fonction non prévue;

b) les cas d'essai et leurs résultats doivent être documentés pour une analyse ultérieure et une évaluation indépendante, comme requis par le niveau d'intégrité de sécurité (voir Article 8 de la CEI 61508-1);

c) les résultats documentés de la validation de sécurité du logiciel doivent établir soit (1) la validation effective du logiciel, soit (2) les raisons de la non-validation.

7.8 Modification du logiciel

NOTE Cette phase correspond à la case 10.5 de la Figure 4.

7.8.1 Objectif

L'objectif des exigences du présent paragraphe est d'apporter des corrections, des améliorations ou des adaptations au logiciel validé en s'assurant du maintien de la capabilité systématique du logiciel requise.

7.8.2 Exigences

NOTE Pour la sélection des techniques et mesures appropriées (voir Annexes A et B) visant à mettre en œuvre les exigences du présent article, il convient de tenir compte des propriétés suivantes (voir l'Annexe C pour les recommandations sur l'interprétation des propriétés, et l'Annexe F de la CEI 61508-7 pour les définitions informelles) de la modification du logiciel:

– complétude de la modification par rapport à ses exigences;

– exactitude de la modification par rapport à ses exigences;

– insensibilité à l'introduction d'anomalies de conception intrinsèques;

– évitement d'un comportement indésirable;

– conception vérifiable et pouvant être soumise à essai;

– essais de régression et couverture de vérification.

7.8.2.1 Avant d'effectuer toute modification du logiciel, on doit s'assurer que les procédures de modification du logiciel sont disponibles (voir 7.16 de la CEI 61508-1).

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

NOTE 1 Les paragraphes 7.8.2.1 à 7.8.2.9 s'appliquent principalement aux changements arrivant pendant la phase d'exploitation du logiciel. Il est permis aussi de les appliquer pendant les phases d'intégration de l'électronique programmable et d'installation et mise en service globales (voir 7.13 de la CEI 61508-1).

NOTE 2 La Figure 9 de la CEI 61508-1 montre un exemple de modèle de procédure de modification.

7.8.2.2 Une modification ne doit être lancée que si la demande de modification du logiciel a fait l'objet d'un accord selon les procédures spécifiées pendant la planification de sécurité (voir l'Article 6), qui décrit en détail:

a) les dangers susceptibles d'être rencontrés;

b) la modification proposée;

c) les raisons de la modification.

NOTE La demande de modification peut, par exemple, être motivée par:

- une sécurité fonctionnelle effective inférieure à celle requise par la spécification des exigences de sécurité;
- une anomalie systématique mise en évidence par expérimentation;
- une législation de sécurité nouvelle ou modifiée;
- des modifications apportées à l'EUC ou à son utilisation;
- une modification des exigences globales de sécurité;
- une analyse des performances d'exploitation et de maintenance, indiquant que ces performances sont inférieures à celles prévues;
- des audits de sécurité fonctionnelle systématiques.

7.8.2.3 Une analyse portant sur l'impact de la modification du logiciel proposée sur la sécurité fonctionnelle du système E/E/PE relatif à la sécurité doit être effectuée:

a) pour déterminer si une analyse de danger et de risque est nécessaire ou non;

b) pour déterminer quelles phases du cycle de vie de sécurité du logiciel doivent être répétées.

7.8.2.4 Les résultats de l'analyse d'impact obtenus en 7.8.2.3 doivent être documentés.

7.8.2.5 Toutes les modifications qui ont un impact sur la sécurité fonctionnelle du système E/E/PE relatif à la sécurité doivent donner lieu à un retour à une phase appropriée du cycle de vie de sécurité du logiciel. Toutes les phases ultérieures doivent ensuite être exécutées conformément aux procédures spécifiées pour les phases spécifiques conformément aux exigences de la présente norme. La planification de sécurité (voir l'Article 6) doit détailler toutes les activités ultérieures.

NOTE Il peut s'avérer nécessaire d'effectuer une analyse complète des dangers et des risques, qui peut entraîner la nécessité de modifier les niveaux d'intégrité de sécurité par rapport à ceux spécifiés pour les fonctions de sécurité exécutées par les systèmes E/E/PE relatifs à la sécurité.

7.8.2.6 La planification de sécurité pour la modification d'un logiciel de sécurité doit satisfaire aux exigences de l'Article 6 de la CEI 61508-1. Notamment:

a) identification du personnel et spécification de leurs compétences requises;

b) spécification détaillée de la modification;

c) planification de la vérification;

d) domaine de revalidation et essai de la modification dans les limites requises par le niveau d'intégrité de sécurité.

NOTE L'implication d'experts du domaine peut se révéler importante selon la nature de l'application.

7.8.2.7 La modification doit être effectuée selon la planification.

7.8.2.8 Les détails de toutes les modifications doivent être documentés en faisant référence à:

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

a) la demande de modification/remise à niveau;

b) les résultats de l'analyse d'impact évaluant l'impact de la modification logicielle proposée sur la sécurité fonctionnelle, ainsi que les décisions prises accompagnées de leurs justifications;

c) l'historique de la gestion de configuration logicielle;

d) les écarts par rapport à l'exploitation et aux conditions normales;

e) toutes les informations documentées affectées par l'activité de modification.

7.8.2.9 Les informations (par exemple, un journal) concernant le détail de toutes les modifications doivent être documentées. La documentation doit contenir la revérification et la revalidation des données ainsi que des résultats.

7.8.2.10 L'évaluation de l'activité de modification ou de rénovation requise doit dépendre des résultats de l'analyse d'impact et de la capabilité systématique du logiciel.

7.9 Vérification du logiciel

7.9.1 Objectif

L'objectif des exigences du présent paragraphe est, dans les limites requises par le niveau d'intégrité de sécurité, de soumettre à l'essai et d'évaluer les sorties d'une phase donnée du cycle de vie de sécurité du logiciel pour assurer la conformité et la cohérence par rapport aux entrées de cette phase.

NOTE 1 Le présent paragraphe prend en compte les aspects génériques de la vérification qui sont communs à plusieurs phases du cycle de vie de sécurité. Il n'implique aucune exigence supplémentaire pour les éléments d'essai de vérification spécifiés en 7.4.7 (essai des modules logiciels), 7.4.8 (intégration du logiciel) et 7.5 (intégration de l'électronique programmable), dans la mesure où ceux-ci constituent en eux-mêmes des activités de vérification. De plus, aucune vérification n'est requise dans le présent paragraphe, autre que la validation du logiciel (voir 7.7) qui, dans la présente norme, constitue le processus consistant à démontrer la conformité à la spécification des exigences de sécurité. Le contrôle du caractère correct de la spécification des exigences de sécurité proprement dit est effectué par des experts du domaine.

NOTE 2 Selon l'architecture du logiciel, la responsabilité de l'activité de vérification peut être partagée entre toutes les organisations impliquées dans le développement et la modification du logiciel.

7.9.2 Exigences

NOTE Pour la sélection des techniques et mesures appropriées (voir Annexes A et B) visant à mettre en œuvre les exigences du présent article, il convient de tenir compte des propriétés suivantes (voir l'Annexe C pour les recommandations sur l'interprétation des propriétés, et l'Annexe F de la CEI 61508-7 pour les définitions informelles) de la vérification des données:

– complétude de la vérification par rapport à la phase précédente;

– exactitude de la vérification par rapport à la phase précédente (réalisation satisfaisante);

– répétabilité;

– configuration de vérification définie avec précision.

7.9.2.1 La vérification du logiciel doit être planifiée (voir 7.3) parallèlement au développement, pour chaque phase du cycle de vie de sécurité du logiciel, et doit être documentée.

7.9.2.2 La planification de la vérification du logiciel doit faire référence aux critères, techniques et outils à utiliser au cours des activités de vérification et doit comprendre:

a) l'évaluation des exigences d'intégrité de sécurité;

b) la sélection et la documentation des stratégies, activités et techniques de vérification;

c) la sélection et l'utilisation des outils de vérification (simulateur d'essai, logiciel d'essai spécial, simulateurs d'entrées/sorties, etc.);

d) l'évaluation des résultats de la vérification;

e) les actions correctives à entreprendre.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

7.9.2.3 La vérification du logiciel doit être effectuée selon la planification.

NOTE La sélection des techniques, les mesures de vérification et le degré d'indépendance des activités de vérification dépendent d'un certain nombre de facteurs et peuvent être spécifiés dans les normes de secteur d'application. Ces facteurs peuvent, par exemple, inclure:

- la taille du projet;
- le degré de complexité;
- le degré de nouveauté de la conception;
- le degré de nouveauté technologique.

7.9.2.4 Les preuves doivent être documentées afin de montrer que la phase faisant l'objet d'une vérification a donné des résultats satisfaisants à tous égards.

7.9.2.5 Après chaque vérification, la documentation de vérification doit comprendre:

a) l'identification des éléments à vérifier;

b) l'identification des informations par rapport auxquelles la vérification a été effectuée;

NOTE 1 Les informations par rapport auxquelles la vérification a été effectuée incluent, sans toutefois s'y limiter, les entrées de la phase de cycle de vie antérieure, les normes de conception, les règles de codage et les outils utilisés.

c) les non-conformités.

NOTE 2 Des exemples de non-conformités comprennent les modules logiciels, les structures de données et les algorithmes peu adaptés au problème.

7.9.2.6 Toutes les informations essentielles de la phase N du cycle de vie de sécurité du logiciel nécessaires à l'exécution correcte de la phase N+1 suivante doivent être disponibles et doivent être vérifiées. Les sorties de la phase N comprennent:

a) l'adéquation de la spécification, de la conception ou du code de la phase N en ce qui concerne:
   1) la fonctionnalité;
   2) l'intégrité de sécurité, les performances et autres exigences de la planification de sécurité (voir Article 6);
   3) la lisibilité par l'équipe de développement;
   4) la testabilité pour vérification ultérieure;
   5) une modification sûre permettant une évolution ultérieure;
b) l'adéquation de la planification de la validation et/ou des essais spécifiés pour la phase N pour spécifier et décrire la conception de la phase N;
c) vérifier les incompatibilités entre:
   1) les essais spécifiés pendant la phase N et les essais spécifiés dans la phase précédente N–1;
   2) les sorties de la phase N.

7.9.2.7 Selon le choix du cycle de vie de développement du logiciel (voir 7.1), les activités de vérification suivantes doivent être effectuées:

a) vérification des exigences pour la sécurité du logiciel;

b) vérification de l'architecture du logiciel;

c) vérification de la conception du système logiciel;

d) vérification de la conception des modules logiciels;

e) vérification du code;

f) vérification des données;

g) vérification des performances de synchronisation;

h) essai des modules logiciels (voir 7.4.7);

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

i) essai d'intégration du logiciel (voir 7.4.8);

j) essai d'intégration de l'électronique programmable (voir 7.5);

k) validation de sécurité du logiciel (voir 7.7).

NOTE Pour les exigences a) à g), voir ci-dessous.

7.9.2.8 Vérification des exigences pour la sécurité du logiciel: une fois que les exigences pour la sécurité du logiciel ont été spécifiées, et avant que la phase suivante de conception et développement du logiciel ne soit lancée, la vérification doit:

a) prendre en compte le fait que la spécification des exigences pour la sécurité du logiciel satisfait de façon appropriée à la spécification des exigences de sécurité du système E/E/PE (voir 7.10 de la CEI 61508-1 et 7.2 de la CEI 61508-2) en ce qui concerne la fonctionnalité, l'intégrité de sécurité, les performances et les autres exigences de la planification de sécurité éventuelles;

b) prendre en compte le fait que le plan de validation de la sécurité du logiciel satisfait de façon appropriée à la spécification des exigences pour la sécurité du logiciel;

c) vérifier les incompatibilités entre:

   1) la spécification des exigences pour la sécurité du logiciel et la spécification des exigences de sécurité du système E/E/PE (voir 7.10 de la CEI 61508-1 et 7.2 de la CEI 61508-2);

   2) la spécification des exigences pour la sécurité du logiciel et le plan de validation de la sécurité du logiciel.

7.9.2.9 Vérification de l'architecture du logiciel: après que la conception de l'architecture du logiciel a été établie, la vérification doit:

a) prendre en compte le fait que la conception de l'architecture du logiciel satisfait de façon appropriée à la spécification des exigences pour la sécurité du logiciel;

b) prendre en compte le fait que les essais d'intégration spécifiés dans la conception de l'architecture du logiciel sont adaptés;

c) prendre en compte le fait que les attributs de chaque composant/sous-système principal sont appropriés en matière de:

   1) faisabilité des performances de sécurité requises;

   2) testabilité pour vérification ultérieure;

   3) lisibilité par l'équipe de développement et de vérification;

   4) modification sûre permettant une évolution ultérieure.

d) vérifier les incompatibilités entre:

   1) la conception de l'architecture du logiciel et la spécification des exigences pour la sécurité du logiciel;

   2) la conception de l'architecture du logiciel et ses essais d'intégration;

   3) les essais d'intégration de la conception de l'architecture du logiciel et le plan de validation de la sécurité du logiciel.

7.9.2.10 Vérification de la conception du système logiciel: après que la conception du système logiciel a été spécifiée, la vérification doit:

a) prendre en compte le fait que la conception du système logiciel (voir 7.4.5) satisfait de façon appropriée à la conception de l'architecture du logiciel;

b) prendre en compte le fait que les essais spécifiés de l'intégration du système logiciel (voir 7.4.5) satisfont de façon appropriée à la conception du système logiciel (voir 7.4.5);

c) prendre en compte le fait que les attributs de chaque composant principal de la spécification de la conception du système logiciel (voir 7.4.5) sont appropriés en matière de:

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

1) faisabilité des performances de sécurité requises;
2) testabilité pour vérification ultérieure;
3) lisibilité par l'équipe de développement et de vérification;
4) modification sûre permettant une évolution ultérieure.

NOTE Les essais d'intégration du système logiciel peuvent être spécifiés comme faisant partie des essais d'intégration de l'architecture du logiciel.

d) vérifier les incompatibilités entre:
1) la spécification de conception du système logiciel (voir 7.4.5) et la conception de l'architecture du logiciel;
2) la spécification de conception du système logiciel (voir 7.4.5) et la spécification des essais d'intégration du système logiciel (voir 7.4.5);
3) les essais requis par la spécification des essais d'intégration du système logiciel (voir 7.4.5) et la spécification des essais d'intégration de l'architecture du logiciel (voir 7.4.3).

7.9.2.11 Vérification de la conception des modules logiciels: après que la conception de chaque module logiciel a été spécifiée, la vérification doit:

a) prendre en compte le fait que la spécification de conception des modules logiciels (voir 7.4.5) satisfait de façon appropriée à la spécification de conception du système logiciel (voir 7.4.5);
b) prendre en compte le fait que la spécification des essais pour chaque module logiciel (voir 7.4.5) est adaptée à la spécification de conception des modules logiciels (voir 7.4.5);
c) prendre en compte le fait que les attributs de chaque module logiciel sont appropriés en matière de:
1) faisabilité des performances de sécurité requises (voir spécification des exigences pour la sécurité du logiciel);
2) testabilité pour vérification ultérieure;
3) lisibilité par l'équipe de développement et de vérification;
4) modification sûre permettant une évolution ultérieure.
d) vérifier les incompatibilités entre:
1) la spécification de conception des modules logiciels (voir 7.4.5) et la spécification de conception du système logiciel (voir 7.4.5);
2) (pour chaque module logiciel) la spécification de conception des modules logiciels (voir 7.4.5) et la spécification des essais des modules logiciels (voir 7.4.5);
3) la spécification des essais des modules logiciels (voir 7.4.5) et la spécification des essais d'intégration du système logiciel (voir 7.4.5).

7.9.2.12 Vérification du code: le code source doit être vérifié par des méthodes statiques permettant d'assurer la conformité à la spécification de conception du module logiciel (voir 7.4.5), aux règles de codage requises (voir 7.4.4) et au plan de validation de la sécurité du logiciel.

NOTE Au cours des premières phases du cycle de vie de sécurité du logiciel, la vérification est statique (par exemple, inspection, revue, preuve formelle, etc.). La vérification du code comprend des techniques telles que les inspections du logiciel et les lectures croisées. La combinaison des résultats de la vérification du code et de l'essai des modules logiciels permet d'assurer que chaque module logiciel satisfait à la spécification qui lui est associée. A partir de là, les essais deviennent le principal moyen de vérification.

7.9.2.13 Vérification des données.

a) Les structures de données doivent être vérifiées.
b) Les données d'application doivent être vérifiées en ce qui concerne:
1) la cohérence avec les structures de données;

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

2) la complétude par rapport aux exigences d'application;

3) la compatibilité avec le logiciel système de base (par exemple, séquence d'exécution, durée d'exécution, etc.); et

4) l'exactitude des valeurs de données.

c) Tous les paramètres d'exploitation doivent être vérifiés en fonction des exigences d'application.

d) Toutes les interfaces de l'installation et leur logiciel associé (c'est-à-dire capteurs et actionneurs et interfaces hors-ligne: voir 7.2.2.11) doivent être vérifiées pour:

1) la détection des défaillances d'interface prévisibles;

2) la tolérance aux défaillances d'interface prévisibles.

e) Toutes les interfaces de communication et leur logiciel associé doivent être vérifiés pour assurer qu'ils ont un niveau approprié de:

1) détection des défaillances;

2) protection contre l'altération;

3) validation des données.

7.9.2.14 Vérification des performances de synchronisation: la prévisibilité du comportement dans le domaine temporel doit être vérifiée.

NOTE Le comportement en synchronisation peut comprendre: les performances, les ressources, le temps de réponse, la durée d'exécution la plus défavorable, l'emballement, l'absence d'interblocage et le système d'exécution.

8 Evaluation de la sécurité fonctionnelle

NOTE Pour la sélection des techniques et mesures appropriées (voir Annexes A et B) visant à mettre en œuvre les exigences du présent article, il convient de tenir compte des propriétés suivantes (voir l'Annexe C pour les recommandations sur l'interprétation des propriétés, et l'Annexe F de la CEI 61508-7 pour les définitions informelles) de l'évaluation de la sécurité fonctionnelle:

– complétude de l'évaluation de la sécurité fonctionnelle par rapport à la présente norme;

– exactitude de l'évaluation de la sécurité fonctionnelle par rapport aux spécifications de conception (réalisation satisfaisante);

– résolution traçable de tous les problèmes identifiés;

– aptitude à modifier l'évaluation de la sécurité fonctionnelle après modification sans nécessité d'un remaniement étendu de l'évaluation;

– répétabilité;

– rapidité d'exécution;

– configuration définie avec précision.

8.1 L'objectif et les exigences de l'Article 8 de la CEI 61508-1 s'appliquent à l'évaluation du logiciel de sécurité.

8.2 Sauf indication contraire dans les normes internationales liées à des secteurs d'application, le niveau minimal d'indépendance du personnel chargé de l'évaluation de la sécurité fonctionnelle doit être celui spécifié à l'Article 8 de la CEI 61508-1.

8.3 Les résultats des activités mentionnées dans le Tableau A.10 peuvent être utilisés pour l'évaluation de la sécurité fonctionnelle.

NOTE Sélectionner des techniques dans les Annexes A et B n'est pas suffisant pour garantir que l'intégrité de sécurité requise est atteinte (voir 7.1.2.7). Il convient que l'évaluateur considère également:

- la cohérence et la complémentarité des méthodes, langages et outils choisis pour l'ensemble du cycle de développement;
- le fait de déterminer si les développeurs utilisent des méthodes, langages et outils qu'ils comprennent parfaitement;
- le fait de déterminer si les méthodes, langages et outils sont bien adaptés aux problèmes spécifiques rencontrés pendant le développement.

--`,``,,`````````,,``,`,,``,`,`-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Annexe A
(normative)

Guide de sélection de techniques et mesures

Certains paragraphes de la présente norme sont associés à un tableau. Par exemple, le paragraphe 7.2 (spécification des exigences pour la sécurité du logiciel) est associé au Tableau A.1. L'Annexe B comporte des tableaux plus détaillés qui reprennent certaines entrées des tableaux de l'Annexe A. Par exemple, le Tableau B.2 reprend le thème de l'essai et de l'analyse dynamique du Tableau A.5.

Voir la CEI 61508-7 pour une présentation des techniques et mesures spécifiques mentionnées dans les Annexes A et B.

Chaque technique ou mesure mentionnée dans les tableaux est assortie d'une recommandation pour les niveaux d'intégrité de sécurité 1 à 4. Ces recommandations se présentent de la manière suivante:

HR	la technique ou mesure est hautement recommandée pour ce niveau d'intégrité de sécurité. Si cette technique ou mesure n'est pas utilisée, il convient alors de justifier ce fait avec référence à l'Annexe C pendant la planification de sécurité et après accord de l'évaluateur.
R	la technique ou mesure est recommandée pour ce niveau d'intégrité de sécurité. Il s'agit d'une recommandation de niveau moins élevé qu'une recommandation HR.
---	l'utilisation de la technique ou mesure n'est ni recommandée ni déconseillée.
NR	la technique ou mesure n'est absolument pas recommandée pour ce niveau d'intégrité de sécurité. Si cette technique ou mesure est utilisée, il convient alors de justifier ce fait avec référence à l'Annexe C pendant la planification de sécurité et après accord de l'évaluateur.

Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité. Des techniques/mesures équivalentes ou de remplacement sont indiquées à l'aide d'une lettre placée à la suite du numéro. Une seule des techniques/mesures équivalentes ou de remplacement doit être satisfaite.

D'autres mesures et techniques peuvent être appliquées à condition de satisfaire aux exigences et objectifs. Voir l'Annexe C pour les recommandations sur la sélection des techniques.

La classification des techniques et des mesures est liée au concept d'*efficacité* utilisé dans la CEI 61508-2. Lorsque tous les autres facteurs sont égaux, les techniques classées HR sont plus efficaces que celles classées R soit pour empêcher l'introduction d'anomalies systématiques lors du développement du logiciel, soit (dans le cas de l'architecture du logiciel) pour maîtriser des anomalies résiduelles du logiciel révélées en cours d'exécution.

En raison du nombre élevé de facteurs qui affectent la capabilité systématique du logiciel, il n'est pas possible d'élaborer un algorithme combinant les techniques et mesures qui sont correctes pour toute application donnée. L'Annexe C donne des recommandations sur une justification de la sélection de techniques spécifiques pour atteindre la capabilité systématique du logiciel.

Pour une application particulière, la combinaison appropriée de techniques ou de mesures doit être établie pendant la planification de sécurité en sélectionnant les techniques ou mesures appropriées, sauf si d'autres exigences figurent dans la note attenante au tableau.

La CEI 61508-6 fournit des recommandations initiales sous la forme de deux exemples d'un travail consacrés à l'interprétation des tableaux.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Tableau A.1 – Spécification des exigences pour la sécurité du logiciel

(Voir 7.2)

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1a	Méthodes semi-formelles	Tableau B.7	R	R	HR	HR
1b	Méthodes formelles	B.2.2, C.2.4	---	R	R	HR
2	Traçabilité ascendante entre les exigences pour la sécurité du système et les exigences pour la sécurité du logiciel	C.2.11	R	R	HR	HR
3	Traçabilité descendante entre les exigences de sécurité et les besoins de sécurité perçus	C.2.11	R	R	HR	HR
4	Outils de spécification assistés par ordinateur venant à l'appui des techniques/mesures appropriées susmentionnées	B.2.4	R	R	HR	HR

NOTE 1 La spécification des exigences pour la sécurité du logiciel exige toujours une description du problème en langage naturel, ainsi que toute notation mathématique nécessaire qui reflète l'application.

NOTE 2 Le tableau reflète les exigences supplémentaires concernant la spécification claire et précise des exigences pour la sécurité du logiciel.

NOTE 3 Voir Tableau C.1.

NOTE 4 Les références (informatives et non pas normatives) "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité. Des techniques/mesures équivalentes ou de remplacement sont indiquées à l'aide d'une lettre placée à la suite du numéro. Il convient de satisfaire normalement à une seule des techniques/mesures équivalentes ou de remplacement. Il convient de justifier la sélection d'une technique de remplacement conformément aux propriétés, spécifiées à l'Annexe C, souhaitées pour l'application particulière.

Tableau A.2 – Conception et développement du logiciel – conception de l'architecture du logiciel

(voir 7.4.3)

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
	Caractéristique d'architecture et de conception					
1	Détection d'anomalie	C.3.1	---	R	HR	HR
2	Codes de détection d'erreurs	C.3.2	R	R	R	HR
3a	Programmation d'assertion des défaillances	C.3.3	R	R	R	HR
3b	Techniques de surveillance diversifiées (avec indépendance entre le moniteur et la fonction surveillée avec le même ordinateur)	C.3.4	---	R	R	----
3c	Techniques de surveillance diversifiées (avec séparation entre l'ordinateur de surveillance et l'ordinateur surveillé)	C.3.4	---	R	R	HR
3d	Redondance diversifiée, implémentant la même spécification des exigences de sécurité du logiciel	C.3.5	---	---	---	R
3e	Redondance diversifiée sur le plan fonctionnel, implémentant une spécification des exigences de sécurité du logiciel différente	C.3.5	---	---	R	HR
3f	Récupération par régression	C.3.6	R	R	---	NR
3g	Conception de logiciel sans état (ou conception avec état limité)	C.2.12	---	---	R	HR
4a	Nouvelle sollicitation des mécanismes de rétablissement après anomalie	C.3.7	R	R	---	---
4b	Dégradation progressive	C.3.8	R	R	HR	HR

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
5	Intelligence artificielle – correction des anomalies	C.3.9	---	NR	NR	NR
6	Reconfiguration dynamique	C.3.10	---	NR	NR	NR
7	Approche modulaire	Tableau B.9	HR	HR	HR	HR
8	Utilisation de composants logiciels sécurisés/vérifiés (s'ils existent)	C.2.10	R	HR	HR	HR
9	Traçabilité ascendante entre la spécification des exigences pour la sécurité du logiciel et l'architecture du logiciel	C.2.11	R	R	HR	HR
10	Traçabilité descendante entre la spécification des exigences pour la sécurité du logiciel et l'architecture du logiciel	C.2.11	R	R	HR	HR
11a	Méthodes diagrammatiques structurées **	C.2.1	HR	HR	HR	HR
11b	Méthodes semi-formelles **	Tableau B.7	R	R	HR	HR
11c	Méthodes de conception et d'affinement formelles **	B.2.2, C.2.4	---	R	R	HR
11d	Génération logicielle automatique	C.4.6	R	R	R	R
12	Outils de spécification et de conception assistés par ordinateur	B.2.4	R	R	HR	HR
13a	Comportement cyclique, avec temps de cycle maximal garanti	C.3.11	R	HR	HR	HR
13b	Architecture à déclenchement temporel	C.3.11	R	HR	HR	HR
13c	Dirigé par les événements, avec temps de réponse maximal garanti	C.3.11	R	HR	HR	-
14	Allocation de ressources statiques	C.2.6.3	-	R	HR	HR
15	Synchronisation statique de l'accès aux ressources partagées	C.2.6.3	-	-	R	HR

NOTE 1 Certaines méthodes données au Tableau A.2 concernent les notions de conception, d'autres concernent le mode de représentation de la conception.

NOTE 2 Il convient de prendre en considération les mesures décrites dans ce tableau concernant la tolérance aux anomalies (maîtrise des défaillances) avec les exigences relatives à l'architecture et à la maîtrise des défaillances pour le matériel de l'électronique programmable spécifiées dans la CEI 61508-2.

NOTE 3 Voir Tableau C.2.

NOTE 4 Les mesures du groupe 13 s'appliquent uniquement aux systèmes et logiciels auxquels sont associées des exigences de synchronisation de sécurité.

NOTE 5 Mesure 14. L'utilisation d'objets dynamiques (par exemple, sur la pile d'exécution ou sur un tas) peut imposer des exigences sur la mémoire disponible et également sur la durée d'exécution. L'application de la mesure 14 n'est pas nécessaire en cas d'utilisation d'un compilateur qui assure a) qu'un espace mémoire suffisant est alloué avant exécution à tous les objets et variables dynamiques, ou qui garantit, en cas d'erreur d'allocation de mémoire, qu'un état de sécurité est obtenu; b) que les temps de réponse satisfont aux exigences.

NOTE 6 Mesure 4a. Une nouvelle sollicitation des mécanismes de rétablissement après anomalie est souvent appropriée pour tout niveau SIL. Il convient toutefois de fixer une limite au nombre de nouvelles sollicitations.

NOTE 7 Les références (informatives et non pas normatives) "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité. Des techniques/mesures équivalentes ou de remplacement sont indiquées à l'aide d'une lettre placée à la suite du numéro. Il convient de satisfaire normalement à une seule des techniques/mesures équivalentes ou de remplacement. Il convient de justifier la sélection d'une technique de remplacement conformément aux propriétés, spécifiées à l'Annexe C, souhaitées pour l'application particulière.

** Groupe 11, "Méthodes structurées". Utiliser la mesure 11a uniquement si la mesure 11b ne convient pas au domaine pour SIL 3+4.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Tableau A.3 – Conception et développement du logiciel – outils de support et langage de programmation

(Voir 7.4.4)

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1	Langage de programmation approprié	C.4.5	HR	HR	HR	HR
2	Langage de programmation fortement typé	C.4.1	HR	HR	HR	HR
3	Sous-ensemble de langage	C.4.2	---	---	HR	HR
4a	Outils et traducteurs certifiés	C.4.3	R	HR	HR	HR
4b	Outils et traducteurs: confiance accrue résultant de l'utilisation	C.4.4	HR	HR	HR	HR

NOTE 1 Voir Tableau C.3.

NOTE 2 Les références (informatives et non pas normatives) "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité. Des techniques/mesures équivalentes ou de remplacement sont indiquées à l'aide d'une lettre placée à la suite du numéro. Il convient de satisfaire normalement à une seule des techniques/mesures équivalentes ou de remplacement. Il convient de justifier la sélection d'une technique de remplacement conformément aux propriétés, spécifiées à l'Annexe C, souhaitées pour l'application particulière.

Tableau A.4 – Conception et développement du logiciel – conception détaillée

(Voir 7.4.5 et 7.4.6)

(Comprend la conception du système logiciel, la conception des modules logiciels et le codage)

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1a	Méthodes structurées **	C.2.1	HR	HR	HR	HR
1b	Méthodes semi-formelles **	Tableau B.7	R	HR	HR	HR
1c	Méthodes de conception et d'affinement formelles **	B.2.2, C.2.4	---	R	R	HR
2	Outils de conception assistés par ordinateur	B.3.5	R	R	HR	HR
3	Programmation défensive	C.2.5	---	R	HR	HR
4	Approche modulaire	Tableau B.9	HR	HR	HR	HR
5	Règles de conception et de codage	C.2.6 Tableau B.1	R	HR	HR	HR
6	Programmation structurée	C.2.7	HR	HR	HR	HR
7	Utilisation de composants logiciels sécurisés/vérifiés (s'ils existent)	C.2.10	R	HR	HR	HR
8	Traçabilité ascendante entre la spécification des exigences pour la sécurité du logiciel et la conception du logiciel	C.2.11	R	R	HR	HR

NOTE 1 Voir Tableau C.4.

NOTE 2 Le caractère approprié du développement du logiciel OO pour les systèmes relatifs à la sécurité fait toujours l'objet de discussions. Voir l'Annexe G de la CEI 61508-7 pour les recommandations sur l'architecture et la conception orientées objet.

NOTE 3 Les références (informatives et non pas normatives) "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité. Des techniques/mesures équivalentes ou de remplacement sont indiquées à l'aide d'une lettre placée à la suite du numéro. Il convient de satisfaire normalement à une seule des techniques/mesures équivalentes ou de remplacement. Il convient de justifier la sélection d'une technique de remplacement conformément aux propriétés, spécifiées à l'Annexe C, souhaitées pour l'application particulière.

** Groupe 1, "Méthodes structurées". Appliquer la mesure 1a uniquement si la mesure 1b ne convient pas au domaine pour SIL 3+4.

Tableau A.5 – Conception et développement du logiciel – essai et intégration des modules logiciels

(Voir 7.4.7 et 7.4.8)

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1	Essai probabiliste	C.5.1	---	R	R	R
2	Analyse dynamique et essai	B.6.5 Tableau B.2	R	HR	HR	HR
3	Enregistrement et analyse des données	C.5.2	HR	HR	HR	HR
4	Essais fonctionnels et boîte noire	B.5.1 B.5.2 Tableau B.3	HR	HR	HR	HR
5	Essais de fonctionnement	Tableau B.6	R	R	HR	HR
6	Essais basés sur le modèle	C.5.27	R	R	HR	HR
7	Essai d'interface	C.5.3	R	R	HR	HR
8	Outils de gestion et d'automatisation des essais	C.4.7	R	HR	HR	HR
9	Traçabilité ascendante entre la spécification de conception du logiciel et les spécifications de modules et d'essai d'intégration	C.2.11	R	R	HR	HR
10	Vérification formelle	C.5.12	---	---	R	R

NOTE 1 Les essais de modules logiciels et d'intégration constituent des activités de vérification (voir Tableau B.9).

NOTE 2 Voir Tableau C.5.

NOTE 3 Technique 9. La vérification formelle peut réduire le nombre et l'étendue des essais de modules et d'intégration requis.

NOTE 4 Les références (informatives et non pas normatives) "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Tableau A.6 – Intégration de l'électronique programmable (matériel et logiciel)

(Voir 7.5)

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1	Essais fonctionnels et boîte noire	B.5.1 B.5.2 Tableau B.3	HR	HR	HR	HR
2	Essais de fonctionnement	Tableau B.6	R	R	HR	HR
3	Traçabilité ascendante entre le système et les exigences de conception du logiciel pour l'intégration du matériel/logiciel et les spécifications d'essai d'intégration du matériel/logiciel	C.2.11	R	R	HR	HR
NOTE 1 L'intégration de l'électronique programmable constitue une activité de vérification (voir Tableau A.9). NOTE 2 Voir Tableau C.6. NOTE 3 Les références (informatives et non pas normatives) "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.						
* Des techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité.						

Tableau A.7 – Validation de sécurité du logiciel

(Voir 7.7)

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1	Essai probabiliste	C.5.1	---	R	R	HR
2	Simulation de processus	C.5.18	R	R	HR	HR
3	Modélisation	Tableau B.5	R	R	HR	HR
4	Essais fonctionnels et boîte noire	B.5.1 B.5.2 Tableau B.3	HR	HR	HR	HR
5	Traçabilité ascendante entre la spécification des exigences pour la sécurité du logiciel et le plan de validation de la sécurité du logiciel	C.2.11	R	R	HR	HR
6	Traçabilité descendante entre le plan de validation de la sécurité du logiciel et la spécification des exigences pour la sécurité du logiciel	C.2.11	R	R	HR	HR
NOTE 1 Voir Tableau C.7. NOTE 2 Les références (informatives et non pas normatives) "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.						
* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité.						

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Tableau A.8 – Modification

(Voir 7.8)

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1	Analyse d'impact	C.5.23	HR	HR	HR	HR
2	Revérification du module logiciel modifié	C.5.23	HR	HR	HR	HR
3	Revérification des modules logiciels affectés	C.5.23	R	HR	HR	HR
4a	Revalidation du système complet	Tableau A .7	---	R	HR	HR
4b	Validation de non-régression	C.5.25	R	HR	HR	HR
5	Gestion de configuration logicielle	C.5.24	HR	HR	HR	HR
6	Enregistrement et analyse des données	C.5.2	HR	HR	HR	HR
7	Traçabilité ascendante entre la spécification des exigences pour la sécurité du logiciel et le plan de modification du logiciel (y compris la revérification et la revalidation)	C.2.11	R	R	HR	HR
8	Traçabilité descendante entre le plan de modification du logiciel (y compris la revérification et la revalidation) et la spécification des exigences pour la sécurité du logiciel	C.2.11	R	R	HR	HR

NOTE 1 Voir Tableau C.8.

NOTE 2 Techniques du groupe 4. L'analyse d'impact constitue une partie nécessaire de la validation de non-régression. Voir la CEI 61508-7.

NOTE 3 Les références (informatives et non pas normatives) "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité. Des techniques/mesures équivalentes ou de remplacement sont indiquées à l'aide d'une lettre placée à la suite du numéro. Il convient de satisfaire normalement à une seule des techniques/mesures équivalentes ou de remplacement. Il convient de justifier la sélection d'une technique de remplacement conformément aux propriétés, spécifiées à l'Annexe C, souhaitées pour l'application particulière.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Tableau A.9 – Vérification du logiciel

(Voir 7.9)

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1	Preuve formelle	C.5.12	---	R	R	HR
2	Animation de la spécification et de la conception	C.5.26	R	R	R	R
3	Analyse statique	B.6.4 Tableau B.8	R	HR	HR	HR
4	Analyse dynamique et essai	B.6.5 Tableau B.2	R	HR	HR	HR
5	Traçabilité ascendante entre la spécification de conception du logiciel et le plan de vérification du logiciel (y compris la vérification des données)	C.2.11	R	R	HR	HR
6	Traçabilité descendante entre le plan de vérification du logiciel (y compris la vérification des données) et la spécification de conception du logiciel	C.2.11	R	R	HR	HR
7	Analyse numérique hors ligne	C.2.13	R	R	HR	HR
Essai des modules logiciels et intégration		Voir le Tableau A.5				
Essai d'intégration de l'électronique programmable		Voir le Tableau A.6				
Essai du système logiciel (validation)		Voir le Tableau A.7				

NOTE 1 Pour des raisons pratiques, toutes les activités de vérification ont été rassemblées dans le présent tableau. Toutefois, cela n'implique aucune exigence supplémentaire pour les éléments d'essai dynamique de vérification du Tableau A.5 et du Tableau A.6 qui constituent en eux-mêmes des activités de vérification. De plus, aucun essai de vérification n'est requis dans ce tableau, en plus de la validation du logiciel (voir Tableau B.7) qui, dans la présente norme, constitue la démonstration de la conformité à la spécification des exigences de sécurité (vérification de bout en bout).

NOTE 2 La vérification couvre la CEI 61508-1, la CEI 61508-2 et la CEI 61508-3. La première vérification du système relatif à la sécurité est donc réalisée par rapport aux spécifications du niveau précédent du système.

NOTE 3 Au cours des premières phases du cycle de vie de sécurité du logiciel, la vérification est statique, par exemple, inspection, revue, preuve formelle. Lorsque le code est produit, il est possible de réaliser un essai dynamique. La combinaison des deux types d'informations est exigée pour la vérification. Par exemple, la vérification du code d'un module logiciel par des moyens statiques comprend les techniques d'inspection du logiciel, de lecture croisée, d'analyse statique, de preuve formelle. La vérification du code par des moyens dynamiques comprend l'essai fonctionnel, l'essai boîte blanche et l'essai statistique. La combinaison des deux types de preuves fournit l'assurance que chaque module logiciel satisfait à sa spécification associée.

NOTE 4 Voir Tableau C.9.

NOTE 5 Les références (informatives et non pas normatives) "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Tableau A.10 – Evaluation de la sécurité fonctionnelle

(voir Article 8)

Evaluation/Technique*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1	Listes de contrôle	B.2.5	R	R	R	R
2	Tables de décision/de vérité	C.6.1	R	R	R	R
3	Analyse des défaillances	Tableau B.4	R	R	HR	HR
4	Analyse des défaillances de cause commune d'un logiciel diversifié (si un logiciel diversifié est effectivement utilisé)	C.6.3	---	R	HR	HR
5	Diagramme de blocs de fiabilité	C.6.4	R	R	R	R
6	Traçabilité ascendante entre les exigences de l'Article 8 et le plan d'évaluation de la sécurité fonctionnelle du logiciel	C.2.11	R	R	HR	HR

NOTE 1 Voir Tableau C.10.

NOTE 2 Les références (informatives et non pas normatives) "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Annexe B
(informative)

Tableaux détaillés

Tableau B.1 – Règles de conception et de codage

(Référencées dans le Tableau A.4)

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1	Utilisation de règles de codage pour réduire la probabilité d'erreurs	C.2.6.2	HR	HR	HR	HR
2	Pas d'objets dynamiques	C.2.6.3	R	HR	HR	HR
3a	Pas de variables dynamiques	C.2.6.3	---	R	HR	HR
3b	Contrôle en ligne pendant la création de variables dynamiques	C.2.6.4	---	R	HR	HR
4	Utilisation limitée des interruptions	C.2.6.5	R	R	HR	HR
5	Utilisation limitée des pointeurs	C.2.6.6	---	R	HR	HR
6	Utilisation limitée de la récursion	C.2.6.7	---	R	HR	HR
7	Pas de branchements inconditionnels dans les programmes en langages de haut niveau	C.2.6.2	R	HR	HR	HR
8	Pas de conversion de type automatique	C.2.6.2	R	HR	HR	HR

NOTE 1 Mesures 2, 3a et 5. L'utilisation d'objets dynamiques (par exemple, sur la pile d'exécution (stack) ou sur un tas) peut imposer des exigences sur la mémoire disponible et également sur la durée d'exécution. L'application des mesures 2, 3a et 5 n'est pas nécessaire en cas d'utilisation d'un compilateur qui assure a) qu'un espace mémoire suffisant est alloué avant exécution à tous les objets et variables dynamiques, ou qui garantit, en cas d'erreur d'allocation de mémoire, qu'un état de sécurité est obtenu; b) que les temps de réponse satisfont aux exigences.

NOTE 2 Voir Tableau C.11.

NOTE 3 Les références (informatives et non pas normatives) "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité. Des techniques/mesures équivalentes ou de remplacement sont indiquées à l'aide d'une lettre placée à la suite du numéro. Il convient de satisfaire normalement à une seule des techniques/mesures équivalentes ou de remplacement. Il convient de justifier la sélection d'une technique de remplacement conformément aux propriétés, spécifiées à l'Annexe C, souhaitées pour l'application particulière.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Tableau B.2 – Analyse dynamique et essai

(Référencés dans les Tableaux A.5 et A.9)

	Technique/Mesure*	réf.	SIL 1	SIL 2	SIL 3	SIL 4
1	Exécution de cas d'essai à partir de l'analyse des valeurs aux limites	C.5.4	R	HR	HR	HR
2	Exécution de cas d'essai à partir de l'estimation des erreurs	C.5.5	R	R	R	R
3	Exécution de cas d'essai à partir de l'implantation d'erreurs	C.5.6	---	R	R	R
4	Exécution de cas d'essai à partir de la génération de cas d'essai basés sur les modèles	C.5.27	R	R	HR	HR
5	Modélisation du fonctionnement	C.5.20	R	R	R	HR
6	Classes d'équivalence et essai des partitions d'entrée	C.5.7	R	R	R	HR
7a	Couverture d'essais structurels (points d'entrée) à 100 %**	C.5.8	HR	HR	HR	HR
7b	Couverture d'essais structurels (énoncés) à 100 % **	C.5.8	R	HR	HR	HR
7c	Couverture d'essais structurels (branchements) à 100 %**	C.5.8	R	R	HR	HR
7d	Couverture d'essais structurels (conditions, MC/DC) à 100 %**	C.5.8	R	R	R	HR

NOTE 1 L'analyse des cas d'essai se fait au niveau du sous-système et est basée sur la spécification et/ou la spécification et le code.

NOTE 2 Voir Tableau C.12.

NOTE 3 Les références (informatives et non pas normatives) "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité.

** Lorsqu'une couverture d'essais à 100 % ne peut être réalisée (par exemple, couverture des énoncés du code défensif), il convient d'en fournir les raisons de manière appropriée.

Tableau B.3 – Essais fonctionnels et boîte noire

(Référencés dans les Tableaux A.5, A.6 et A.7)

	Technique/Mesure*	réf.	SIL 1	SIL 2	SIL 3	SIL 4
1	Exécution de cas d'essai à partir de diagrammes cause/conséquence	B.6.6.2	---	---	R	R
2	Exécution de cas d'essai à partir de la génération de cas d'essai basés sur les modèles	C.5.27	R	R	HR	HR
3	Prototypage/animation	C.5.17	---	---	R	R
4	Classes d'équivalence et essai des partitions d'entrée, y compris l'analyse des valeurs aux limites	C.5.7 C.5.4	R	HR	HR	HR
5	Simulation de processus	C.5.18	R	R	R	R

NOTE 1 L'analyse des cas d'essai est effectuée au niveau du système logiciel et est basée uniquement sur la spécification.

NOTE 2 La complétude de la simulation dépend du niveau d'intégrité de sécurité, de la complexité et de l'application.

NOTE 3 Voir Tableau C.13.

NOTE 4 Les références (informatives et non pas normatives) "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Tableau B.4 – Analyse de défaillance

(Référencée dans le Tableau A.10)

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1a	Diagrammes cause/conséquence	B.6.6.2	R	R	R	R
1b	Analyse par arbre d'événement	B.6.6.3	R	R	R	R
2	Analyse par arbre de panne	B.6.6.5	R	R	R	R
3	Analyse des défaillances fonctionnelles du logiciel	B.6.6.4	R	R	R	R

NOTE 1 Il convient d'effectuer une analyse de danger préliminaire afin de classer le logiciel par rapport au niveau d'intégrité de sécurité le plus approprié.

NOTE 2 Voir Tableau C.14.

NOTE 3 Les références (informatives et non pas normatives) "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité. Des techniques/mesures équivalentes ou de remplacement sont indiquées à l'aide d'une lettre placée à la suite du numéro. Il convient de satisfaire normalement à une seule des techniques/mesures équivalentes ou de remplacement. Il convient de justifier la sélection d'une technique de remplacement conformément aux propriétés, spécifiées à l'Annexe C, souhaitées pour l'application particulière.

Tableau B.5 – Modélisation

(Référencé dans le Tableau A.7)

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1	Diagrammes de flux de données	C.2.2	R	R	R	R
2a	Automate à états finis	B.2.3.2	---	R	HR	HR
2b	Méthodes formelles	B.2.2, C.2.4	---	R	R	HR
2c	Réseaux de Pétri temporels	B.2.3.3	---	R	HR	HR
3	Modélisation du fonctionnement	C.5.20	R	HR	HR	HR
4	Prototypage/animation	C.5.17	R	R	R	R
5	Diagrammes de structures	C.2.3	R	R	R	HR

NOTE 1 Si une technique spécifique n'est pas mentionnée dans le tableau, il convient de ne pas la considérer comme exclue. Il convient qu'elle soit conforme à la présente norme.

NOTE 2 La quantification des probabilités n'est pas nécessaire.

NOTE 3 Voir Tableau C.15.

NOTE 4 Les références (informatives et non pas normatives) "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité. Des techniques/mesures équivalentes ou de remplacement sont indiquées à l'aide d'une lettre placée à la suite du numéro. Il convient de satisfaire normalement à une seule des techniques/mesures équivalentes ou de remplacement. Il convient de justifier la sélection d'une technique de remplacement conformément aux propriétés, spécifiées à l'Annexe C, souhaitées pour l'application particulière.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Tableau B.6 – Essais de fonctionnement

(Référencés dans les Tableaux A.5 et A.6)

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1	Essais d'avalanche/de stress	C.5.21	R	R	HR	HR
2	Temps de réponse et contraintes mémoire	C.5.22	HR	HR	HR	HR
3	Exigences relatives au fonctionnement	C.5.19	HR	HR	HR	HR

NOTE 1 Voir Tableau C.16.

NOTE 2 Les références (informatives et non pas normatives) "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité.

Tableau B.7 – Méthodes semi-formelles

(Référencées dans les Tableaux A.1, A.2 et A.4)

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1	Diagrammes de blocs logiques/fonctionnels	Voir Note 1 ci-dessous	R	R	HR	HR
2	Diagrammes de séquence	Voir Note 1 ci-dessous	R	R	HR	HR
3	Diagrammes de flux de données	C.2.2	R	R	R	R
4a	Automates finis/diagrammes de changement d'états	B.2.3.2	R	R	HR	HR
4b	Réseaux de Pétri temporels	B.2.3.3	R	R	HR	HR
5	Modèles de données entité-relation-attribut	B.2.4.4	R	R	R	R
6	Diagrammes de séquences de messages	C.2.14	R	R	R	R
7	Tables de décision/de vérité	C.6.1	R	R	HR	HR
8	UML (langage de modélisation unifié)	C.3.12	R	R	R	R

NOTE 1 Les diagrammes de blocs logiques/fonctionnels et les diagrammes de séquence sont décrits dans la CEI 61131-3.

NOTE 2 Voir Tableau C.17.

NOTE 3 Les références "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité. Des techniques/mesures équivalentes ou de remplacement sont indiquées à l'aide d'une lettre placée à la suite du numéro. Il convient de satisfaire normalement à une seule des techniques/mesures équivalentes ou de remplacement. Il convient de justifier la sélection d'une technique de remplacement conformément aux propriétés, spécifiées à l'Annexe C, souhaitées pour l'application particulière.

--`,``,,`````````,,``,`,,``,`,`-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Tableau B.8 – Analyse statique

(Référencée dans le Tableau A.9)

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1	Analyse des valeurs aux limites	C.5.4	R	R	HR	HR
2	Listes de contrôle	B.2.5	R	R	R	R
3	Analyse du flux de commande	C.5.9	R	HR	HR	HR
4	Analyse du flux de données	C.5.10	R	HR	HR	HR
5	Estimation des erreurs	C.5.5	R	R	R	R
6a	Inspections formelles, y compris les critères spécifiques	C.5.14	R	R	HR	HR
6b	Lectures croisées (logiciel)	C.5.15	R	R	R	R
7	Exécution symbolique	C.5.11	---	---	R	R
8	Revue de conception	C.5.16	HR	HR	HR	HR
9	Analyse statique du comportement d'erreur d'exécution	B.2.2, C.2.4	R	R	R	HR
10	Analyse de la durée d'exécution la plus défavorable	C.5.20	R	R	R	R

NOTE 1 Voir Tableau C.18.

NOTE 2 Les références "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité. Des techniques/mesures équivalentes ou de remplacement sont indiquées à l'aide d'une lettre placée à la suite du numéro. Il convient de satisfaire normalement à une seule des techniques/mesures équivalentes ou de remplacement. Il convient de justifier la sélection d'une technique de remplacement conformément aux propriétés, spécifiées à l'Annexe C, souhaitées pour l'application particulière.

Tableau B.9 – Approche modulaire

(Référencée dans le Tableau A.4)

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1	Limitation de la taille des modules logiciels	C.2.9	HR	HR	HR	HR
2	Maîtrise de la complexité du logiciel	C.5.13	R	R	HR	HR
3	Masquage/encapsulation des informations	C.2.8	R	HR	HR	HR
4	Limitation du nombre de paramètres / nombre fixe de paramètres de sous-programme	C.2.9	R	R	R	R
5	Un point d'entrée/un point de sortie dans les sous-programmes et les fonctions	C.2.9	HR	HR	HR	HR
6	Interface totalement définie	C.2.9	HR	HR	HR	HR

NOTE 1 Voir Tableau C.19.

NOTE 2 Les références "B.x.x.x", "C.x.x.x" dans la colonne 3 (réf.) désignent des descriptions détaillées de techniques/mesures données dans les Annexes B et C de la CEI 61508-7.

* Les techniques/mesures appropriées doivent être sélectionnées en fonction du niveau d'intégrité de sécurité. Aucune de ces techniques n'est vraisemblablement suffisante à elle seule. Toutes les techniques appropriées doivent être prises en considération.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Annexe C
(informative)

Propriétés relatives à la capabilité systématique du logiciel

C.1 Introduction

En raison du nombre élevé de facteurs qui affectent la capabilité systématique du logiciel, il n'est pas possible d'élaborer un algorithme combinant les techniques et mesures qui sont correctes pour toute application donnée. La présente Annexe C a pour objet:

– de donner des recommandations sur la sélection de techniques spécifiques définies dans les Annexes A et B, pour atteindre la capabilité systématique du logiciel;
– de présenter succinctement les motifs de justification de l'emploi de techniques non énumérées de manière explicite dans les Annexes A et B.

L'Annexe C complète les tableaux des Annexes A et B.

C.1.1 Structure de l'Annexe C, en relation avec les Annexes A et B

Les sorties de chaque phase du cycle de vie de sécurité du logiciel sont définies dans le Tableau 1. Soit, par exemple, la spécification des exigences pour la sécurité du logiciel.

Le Tableau A.1 («Spécification des exigences pour la sécurité du logiciel») de l'Annexe A recommande des techniques spécifiques pour l'élaboration de la spécification des exigences pour la sécurité du logiciel.

Technique/Mesure*		réf.	SIL 1	SIL 2	SIL 3	SIL 4
1a	Méthodes semi-formelles	Tableau B.7	R	R	HR	HR
1b	Méthodes formelles	B.2.2, C.2.4	---	R	R	HR
2	Traçabilité ascendante entre les exigences pour la sécurité du système et les exigences pour la sécurité du logiciel	C.2.11	R	R	HR	HR
3	Traçabilité descendante entre les exigences de sécurité et les besoins de sécurité perçus	C.2.11	R	R	HR	HR
4	Outils de spécification assistés par ordinateur venant à l'appui des techniques/mesures appropriées susmentionnées	B.2.4	R	R	HR	HR

Le Tableau C.1 de l'Annexe C ("Propriétés d'intégrité systématique – Spécification des exigences pour la sécurité du logiciel") indique que la spécification des exigences pour la sécurité du logiciel est caractérisée par les propriétés souhaitées suivantes (définies de manière informelle dans l'Annexe F de la CEI 61508-7):

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Propriétés					
Complétude par rapport aux besoins de sécurité devant être traités par le logiciel	Exactitude par rapport aux besoins de sécurité devant être traités par le logiciel	Insensibilité aux anomalies de spécification intrinsèques, y compris exemption de toute ambiguïté	Intelligibilité des exigences de sécurité	Insensibilité aux perturbations préjudiciables des fonctions non liées à la sécurité avec les besoins de sécurité devant être traités par le logiciel	Aptitude à fournir une base pour la vérification et la validation

Le Tableau C.1 de l'Annexe C classe également, sur une échelle informelle R1/R2/R3, l'efficacité des techniques spécifiques pour l'obtention des propriétés souhaitées.

Technique/ Mesure		Propriétés					
		Complétude par rapport aux besoins de sécurité devant être traités par le logiciel	Exactitude par rapport aux besoins de sécurité devant être traités par le logiciel	Insensibilité aux anomalies de spécification intrinsèques, y compris exemption de toute ambiguïté	Intelligibilité des exigences de sécurité	Insensibilité aux perturbations préjudiciables des fonctions non liées à la sécurité avec les besoins de sécurité devant être traités par le logiciel	Aptitude à fournir une base pour la vérification et la validation
1a	Méthodes semi-formelles	R1 Spécification conviviale pour l'application ou spécifique au domaine et notation utilisée par les experts du domaine	R1 Spécification conviviale pour l'application ou spécifique au domaine et notation utilisée par les experts du domaine R2 Vérification de la spécification selon les critères de couverture	R1 Méthode et notation qui permettent d'éviter ou de détecter les incohérences internes, tout comportement défaillant ou les expressions mathématiquement incohérentes. R2 Vérification de la spécification selon les critères de couverture R3 Vérification de la spécification fondée sur une analyse systématique, et/ou un évitement systématique des types particuliers d'anomalies de spécification intrinsèques	R1 Notation définie limitant toute possibilité de mauvaise compréhension R2 Application des limites de complexité dans la spécification	–	R2 Notation définie réduisant toute ambiguïté de la spécification

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

La confiance pouvant être accordée à la spécification des exigences pour la sécurité du logiciel comme élément de base d'un logiciel de sécurité, dépend du caractère rigoureux des techniques qui ont permis d'obtenir les propriétés souhaitées de la spécification. Le caractère rigoureux d'une technique est classé de manière informelle sur une échelle R1 à R3, où R1 représentent le caractère le moins rigoureux et R3 représente le caractère le plus rigoureux.

R1	sans critères d'acceptation objectifs, ou avec critères d'acceptation objectifs limités. Par exemple, essai boîte noire basé sur le jugement, expériences de terrain.
R2	avec critères d'acceptation objectifs qui donnent un haut niveau de confiance dans le fait que la propriété requise est obtenue (exceptions d'identification et de justification); par exemple, techniques d'essai ou d'analyse avec métriques de couverture, couverture des listes de contrôle.
R3	avec raisonnement objectif et systématique selon lequel la propriété requise est obtenue. Par exemple, preuve formelle, respect démontré des contraintes d'architecture garantissant la propriété.
_	cette technique ne s'applique pas à cette propriété.

Une technique peut réaliser un classement parmi plusieurs classements R1/R2/R3 relatifs à une propriété particulière, selon le niveau de rigueur de la technique.

Technique/ Mesure		Propriétés					
		Complétude par rapport aux besoins de sécurité devant être traités par le logiciel	Exactitude par rapport aux besoins de sécurité devant être traités par le logiciel	Insensibilité aux anomalies de spécification intrinsèques, y compris exemption de toute ambiguïté	Intelligibilité des exigences de sécurité	Insensibilité aux perturbations préjudiciables des fonctions non liées à la sécurité avec les besoins de sécurité devant être traités par le logiciel	Aptitude à fournir une base pour la vérification et la validation
1a	Méthodes semi-formelles				R1 Notation définie limitant toute possibilité de mauvaise compréhension R2 Application des limites de complexité dans la spécification		

Dans cet exemple, une méthode semi-formelle permet d'atteindre une classe de rigueur R1 grâce à une notation réduite qui améliore la précision de l'expression, et atteint par ailleurs une classe de rigueur R2 par réduction supplémentaire de la complexité de la spécification susceptible, dans le cas contraire, de constituer une source de confusion.

C.1.2 Méthode d'utilisation – 1

S'il peut être démontré avec preuve à l'appui, et à des fins de recommandation, que les propriétés souhaitées ont été obtenues dans l'élaboration de la spécification des exigences pour la sécurité du logiciel, il peut alors être justifié de manière raisonnable que la spécification des exigences pour la sécurité du logiciel constitue une base appropriée de développement d'un logiciel dont l'intégrité de sécurité systématique est suffisante.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Le Tableau C.1 de l'Annexe C indique que chacune des techniques décrites dans le Tableau A.1 de l'Annexe A permet d'obtenir généralement, et plus ou moins, une ou plusieurs propriétés définies dans le Tableau C.1 ci-dessus, pertinentes pour la spécification des exigences pour la sécurité du logiciel.

Il est toutefois important de noter que ces recommandations, bien que le Tableau A.1 de l'Annexe A recommande des techniques spécifiques, ne sont pas prescriptives, l'Annexe A indiquant en fait de manière claire que «En raison du nombre élevé de facteurs qui affectent la capabilité systématique du logiciel, il n'est pas possible d'élaborer un algorithme combinant les techniques et mesures qui sont correctes pour toute application donnée ».

Dans la pratique, les techniques d'élaboration de la spécification des exigences pour la sécurité du logiciel sont sélectionnées dans le cadre de plusieurs contraintes pratiques (voir 7.1.2.7), outre les capabilités intrinsèques des techniques. Ces contraintes peuvent comprendre:

– la cohérence et la complémentarité des méthodes, langages et outils choisis pour l'ensemble du cycle de développement;
– le fait de déterminer si les développeurs utilisent des méthodes, langages et outils qu'ils comprennent parfaitement;
– le fait de déterminer si les méthodes, langages et outils sont bien adaptés aux problèmes spécifiques rencontrés pendant le développement.

Le Tableau C.1 peut être utilisé pour comparer l'efficacité relative des techniques spécifiques définies dans le Tableau A.1 de l'Annexe A pour obtenir les propriétés souhaitées du cycle de vie de la spécification des exigences pour la sécurité du logiciel, et intégrer simultanément les contraintes pratiques du projet de développement particulier.

Par exemple, une méthode formelle constitue une meilleure base (R3) de vérification et de validation qu'une méthode semi-formelle (R2), d'autres contraintes liées au projet (par exemple, la disponibilité d'outils de support informatique élaborés, ou l'expressivité très spécialisée d'une notation formelle) pouvant toutefois plaider pour une approche semi-formelle.

Ainsi, les propriétés souhaitées définies dans le Tableau C.1 constituent la base d'une comparaison pratique raisonnée des techniques de remplacement recommandées par le Tableau A.1 de l'Annexe A pour élaborer la spécification des exigences pour la sécurité du logiciel. Plus généralement, l'étude des propriétés souhaitées énumérées dans le tableau correspondant de l'Annexe C permet de sélectionner de manière raisonnée une ou plusieurs techniques parmi les différentes techniques de remplacement recommandées par l'Annexe A pour une phase particulière du cycle de vie.

Il est toutefois important de bien noter que, du fait de la nature du comportement systématique, ces propriétés définies dans l'Annexe C peuvent ne pas être obtenues ou démontrées avec la plus grande rigueur. Elles représentent en revanche des buts à atteindre. Leur réalisation peut parfois nécessiter des compromis à faire entre les différentes propriétés, par exemple, entre conception défensive et simplicité.

Enfin, et outre la définition des critères R1/R2/R3, il est utile, à des fins de recommandation, d'établir un lien informel entre (1) l'ordre croissant des niveaux de rigueur de R1 à R3 et (2) une confiance accrue dans l'exactitude du logiciel. En tant que recommandation générale et informelle, il convient d'atteindre les niveaux de rigueur minimum suivants lorsque l'Annexe A exige les performances correspondant au niveau SIL:

SIL	Rigueur R
1 / 2	R1
3	R2 le cas échéant
4	plus grande rigueur disponible

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

C.1.3 Méthode d'utilisation – 2

Il est également admis, bien que l'Annexe A recommande des techniques spécifiques, d'appliquer d'autres mesures et techniques, à condition que les exigences et objectifs de la phase du cycle de vie aient été satisfaits.

Il a déjà été constaté que de nombreux facteurs affectent la capabilité systématique du logiciel, et qu'il n'est pas possible de donner un algorithme permettant de sélectionner et de combiner les techniques de manière à garantir la réalisation des propriétés souhaitées dans toute application donnée.

Il peut exister plusieurs moyens efficaces d'obtenir les propriétés souhaitées et il convient d'admettre que les développeurs de système sont en mesure de fournir des preuves de remplacement. Les informations fournies dans ces tableaux de l'Annexe C peuvent être utilisées comme base d'argument pondéré permettant de justifier le choix de techniques autres que celles indiquées dans les tableaux de l'Annexe A.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

C.2 Propriétés relatives à l'intégrité systématique

Les indications données dans le présent document et dans la CEI 61508-7 indiquent des techniques spécifiques permettant d'obtenir les propriétés d'intégrité systématique et de produire des preuves probantes. Lorsqu'une méthode ne permet pas de contribuer à la réalisation d'une propriété, ceci est indiqué dans les tableaux suivants par un tiret. Lorsqu'une méthode peut avoir des effets préjudiciables sur certaines propriétés et des effets positifs sur d'autres, une note est fournie dans le tableau correspondant ci-dessous.

Tableau C.1 – Propriétés relatives à l'intégrité systématique – Spécification des exigences pour la sécurité du logiciel

(Voir 7.2. Référencées dans le Tableau A.1)

Technique/Mesure		Propriétés					
		Complétude par rapport aux besoins de sécurité devant être traités par le logiciel	**Exactitude par rapport aux besoins de sécurité devant être traités par le logiciel**	**Insensibilité aux anomalies de spécification intrinsèques, y compris exemption de toute ambiguïté**	**Intelligibilité des exigences de sécurité**	**Insensibilité aux perturbations préjudiciables des fonctions non liées à la sécurité avec les besoins de sécurité devant être traités par le logiciel**	**Aptitude à fournir une base pour la vérification et la validation**
1a	Méthodes semi-formelles	R1 Spécification conviviale pour l'application ou spécifique au domaine et notation utilisée par les experts du domaine	R1 Spécification conviviale pour l'application ou spécifique au domaine et notation utilisée par les experts du domaine R2 Vérification de la spécification selon les critères de couverture	R1 Méthode et notation qui permettent d'éviter ou de détecter les incohérences internes, tout comportement défaillant ou les expressions mathématiquement incohérentes. R2 Vérification de la spécification selon les critères de couverture R3 Vérification de la spécification fondée sur une analyse systématique, et/ou un évitement systématique des types particuliers d'anomalies de spécification intrinsèques	R1 Notation définie limitant toute possibilité de mauvaise compréhension R2 Application de limites de complexité dans la spécification	–	R2 Notation définie réduisant toute ambiguïté de la spécification

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés					
		Complétude par rapport aux besoins de sécurité devant être traités par le logiciel	**Exactitude par rapport aux besoins de sécurité devant être traités par le logiciel**	**Insensibilité aux anomalies de spécification intrinsèques, y compris exemption de toute ambiguïté**	**Intelligibilité des exigences de sécurité**	**Insensibilité aux perturbations préjudiciables des fonctions non liées à la sécurité avec les besoins de sécurité devant être traités par le logiciel**	**Aptitude à fournir une base pour la vérification et la validation**
1b	Méthodes formelles	R1 Spécification conviviale pour l'application ou spécifique au domaine et notation utilisée par les experts du domaine	R1 Spécification conviviale pour l'application ou spécifique au domaine et notation utilisée par les experts du domaine. R2 Vérification de la spécification selon les critères de couverture R3 Garantie de l'exactitude des aspects restreints du comportement	R1 Méthode et notation qui permettent d'éviter ou de détecter les incohérences internes, tout comportement défaillant ou les expressions mathématiquement incohérentes. R2 Vérification de la spécification selon les critères de couverture R3 Vérification de la spécification fondée sur une analyse systématique, et/ou un évitement systématique des types particuliers d'anomalies de spécification intrinsèques	– Note: Peut rendre plus complexe l'obtention de cette propriété si la méthode n'est pas conviviale pour l'application ou spécifique au domaine.	–	R3 Réduit toute ambiguïté de la spécification.
2	Traçabilité ascendante entre la spécification des exigences pour la sécurité du système et les exigences pour la sécurité du logiciel	R1 Confiance dans le fait que la spécification des exigences pour la sécurité du logiciel traite les exigences pour la sécurité du système	–	–	–	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés					
		Complétude par rapport aux besoins de sécurité devant être traités par le logiciel	**Exactitude par rapport aux besoins de sécurité devant être traités par le logiciel**	**Insensibilité aux anomalies de spécification intrinsèques, y compris exemption de toute ambiguïté**	**Intelligibilité des exigences de sécurité**	**Insensibilité aux perturbations préjudiciables des fonctions non liées à la sécurité avec les besoins de sécurité devant être traités par le logiciel**	**Aptitude à fournir une base pour la vérification et la validation**
3	Traçabilité descendante entre la spécification des exigences pour la sécurité du logiciel et les besoins de sécurité perçus	–	R1 Confiance dans le fait que la spécification des exigences pour la sécurité du logiciel ne comporte pas de complexité inutile	–	RI La traçabilité selon les besoins de sécurité de l'EUC améliore l'intelligibilité	R1	R1
4	Outils de spécification assistés par ordinateur venant à l'appui des techniques/mesures appropriées susmentionnées	R1 Encapsulation des connaissances de domaine de l'EUC et de l'environnement du logiciel R2 si la liste de contrôle des problèmes à prendre en considération est définie, justifiée et traitée	R1 Techniques de simulation fonctionnelle R2 Simulation fonctionnelle selon des critères de couverture définis et justifiés	R2 Vérifications sémantique et syntaxique afin de s'assurer que les règles correspondantes sont satisfaites	R1 Animation ou exploration de la spécification	R1 Identification des fonctions de sécurité et des fonctions non liées à la sécurité	R1 Facilite la traçabilité et la couverture R2 Mesure de la traçabilité et de la couverture

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Tableau C.2 – Propriétés relatives à l'intégrité systématique – Conception et développement du logiciel – Conception de l'architecture logicielle

(voir 7.4.3. Référencées dans le Tableau A.2)

Technique/Mesure		Propriétés							
		Complétude par rapport à la spécification des exigences pour le logiciel de sécurité	**Exactitude par rapport à la spécification des exigences pour le logiciel de sécurité**	**Insensibilité aux anomalies de conception intrinsèques**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies**	**Protection contre les défaillances de cause commune dues à des événements externes**
1	Détection des anomalies	–	–	–	Peut rendre plus complexe l'obtention de cette propriété	R1 Surveillance du déroulement de programme logique garantissant la prévisibilité	–	R1 (R2 si des cibles de couverture sont définies, justifiées et atteintes)	R1 ou –
2	Codes de détection d'erreur	–	–	–	– Note: Peut rendre plus complexe l'obtention de cette propriété.	– Note: Peut rendre plus complexe l'obtention de cette propriété.	–	R1 (R2 si des cibles de couverture sont définies, justifiées et atteintes) Efficace pour des domaines d'application spécifiques, par exemple, transmission de données	R1 Efficace pour des domaines d'application spécifiques, par exemple, transmission de données

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés							
		Complétude par rapport à la spécification des exigences pour le logiciel de sécurité	**Exactitude par rapport à la spécification des exigences pour le logiciel de sécurité**	**Insensibilité aux anomalies de conception intrinsèques**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies**	**Protection contre les défaillances de cause commune dues à des événements externes**
3a	Programmation d'assertion des défaillances	_	R2 Les post-assertions peuvent vérifier la conformité aux exigences détaillées	_	R2 Les assertions préalables limitent l'espace d'entrée	R2 Les post-assertions vérifient les sorties prévues / acceptables	R2 Les assertions préalables limitent l'espace d'entrée et de ce fait l'espace d'essai requis	R3 Efficace pour les défaillances ciblées	R3 Efficace pour les défaillances ciblées
3b	Techniques de surveillance diversifiées (avec indépendance entre le moniteur et la fonction surveillée avec le même ordinateur)	_	_	R2 Le moniteur diversifié implémente uniquement les exigences de sécurité minimales	R2 Le moniteur diversifié prévoit une diversité implicite	R2 Le moniteur diversifié implémente, de façon simple, uniquement les exigences de sécurité minimales	R2 Le moniteur diversifié implémente uniquement les exigences de sécurité minimales	R1 (R2 si des cibles de couverture sont définies, justifiées et atteintes)	R1 (R2 si des cibles de couverture sont définies, justifiées et atteintes)
3c	Techniques de surveillance diversifiées (avec séparation entre l'ordinateur de surveillance et l'ordinateur surveillé)	-	_	R2 Le moniteur diversifié implémente uniquement les exigences de sécurité minimales	R2 Le moniteur diversifié prévoit une diversité implicite	R2 Le moniteur diversifié implémente, de façon simple, uniquement les exigences de sécurité minimales	R2 Le moniteur diversifié implémente uniquement les exigences de sécurité minimales	R1 (R2 si des cibles de couverture sont définies, justifiées et atteintes)	R1 (R2 si des cibles de couverture sont définies, justifiées et atteintes)

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés							
		Complétude par rapport à la spécification des exigences pour le logiciel de sécurité	**Exactitude par rapport à la spécification des exigences pour le logiciel de sécurité**	**Insensibilité aux anomalies de conception intrinsèques**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies**	**Protection contre les défaillances de cause commune dues à des événements externes**
3d	Redondance diversifiée, implémentant la même spécification des exigences de sécurité du logiciel	–	–	–	– Note: Peut rendre plus complexe l'obtention de cette propriété si réalisé dans le même logiciel exécutable.	–	–	R1 Si la défaillance d'un programme n'altère pas les autres programmes R2 si des cibles de couverture sont définies, justifiées et atteintes Ne protège pas contre les anomalies de la spécification des exigences	R1 Si la défaillance d'un programme n'altère pas les autres programmes R2 Si des cibles de couverture sont définies, justifiées et atteintes Ne protège pas contre les anomalies de la spécification des exigences
3e	Redondance diversifiée sur le plan fonctionnel, implémentant une spécification des exigences de sécurité du logiciel différente Ceci nécessite généralement que les capteurs fonctionnent selon des principes physiques différents	–	–	R1	– Note: Peut rendre plus complexe l'obtention de cette propriété si réalisé dans le même logiciel exécutable.	–	–	R1 Si la défaillance d'un programme n'altère pas les autres programmes.	R1 Si la défaillance d'un programme n'altère pas les autres programmes. Protège contre les anomalies de spécification

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés							
		Complétude par rapport à la spécification des exigences pour le logiciel de sécurité	**Exactitude par rapport à la spécification des exigences pour le logiciel de sécurité**	**Insensibilité aux anomalies de conception intrinsèques**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies**	**Protection contre les défaillances de cause commune dues à des événements externes**
3f	Récupération par régression	–	–	– Note: Peut rendre plus complexe l'obtention de cette propriété.	–	– Note: Peut rendre plus complexe l'obtention de cette propriété.	–	R2	R1 (R2 si des cibles de couverture sont définies, justifiées et atteintes)
3g	Conception sans état (ou conception avec état limité)	R2 Sous réserve que les exigences de sécurité soient également sans état ou avec état limité	R2 Sous réserve que les exigences de sécurité soient également sans état ou avec état limité	R2 Sous réserve que les exigences de sécurité soient également sans état ou avec état limité	R1 R2 Si des limites sont définies, justifiées et satisfaites concernant le nombre d'états possibles	R1 R2 Si des limites sont définies, justifiées et satisfaites concernant le nombre d'états possibles	R1 R2 Si des cibles sont définies, justifiées et satisfaites pour la vérification / couverture d'essai des états possibles	R1 Si cela génère une conception à restauration automatique R2 Si des cibles sont définies, justifiées et atteintes pour la restauration automatique	R1 Si cela génère une conception à restauration automatique R2 si des cibles sont définies, justifiées et atteintes pour la restauration automatique
4a	Nouvelle sollicitation des mécanismes de rétablissement après anomalie	–	–	–	–	Peut rendre plus complexe l'obtention de cette propriété	–	R1 (R2 si des cibles de couverture sont définies, justifiées et atteintes)	R1 (R2 si des cibles de couverture sont définies, justifiées et atteintes)

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés							
		Complétude par rapport à la spécification des exigences pour le logiciel de sécurité	**Exactitude par rapport à la spécification des exigences pour le logiciel de sécurité**	**Insensibilité aux anomalies de conception intrinsèques**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies**	**Protection contre les défaillances de cause commune dues à des événements externes**
4b	Dégradation progressive	–	–	– Note: Peut rendre plus complexe l'obtention de cette propriété.	–	–	–	R1 R2 si des cibles de couverture sont définies, justifiées et atteintes	R1 R2 si des cibles de couverture sont définies, justifiées et atteintes
5	Intelligence artificielle – correction des anomalies	–	– Note: Peut rendre plus complexe l'obtention de cette propriété.	– Note: Peut rendre plus complexe l'obtention de cette propriété.	– Note: Peut rendre plus complexe l'obtention de cette propriété.	– Note: Peut rendre plus complexe l'obtention de cette propriété.	– Note: Peut rendre plus complexe l'obtention de cette propriété.	–	–
6	Reconfiguration dynamique	–	– Note: Peut rendre plus complexe l'obtention de cette propriété	– Note: Peut rendre plus complexe l'obtention de cette propriété.	– Note: Peut rendre plus complexe l'obtention de cette propriété.	– Note: Peut rendre plus complexe l'obtention de cette propriété.	– Note: Peut rendre plus complexe l'obtention de cette propriété.	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés							
		Complétude par rapport à la spécification des exigences pour le logiciel de sécurité	**Exactitude par rapport à la spécification des exigences pour le logiciel de sécurité**	**Insensibilité aux anomalies de conception intrinsèques**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies**	**Protection contre les défaillances de cause commune dues à des événements externes**
7	Approche modulaire	_	R1 R2 R2 est obtenue si les cibles de modularité sont définies, justifiées et atteintes. Dans le cas contraire, seule R1 est obtenue.	R1 Si l'insensibilité aux types particuliers d'anomalies de conception intrinsèques peut être vérifiée de manière indépendante pour chaque module R3 Si l'insensibilité aux types particuliers d'anomalies de conception intrinsèques peut être prise en charge par un raisonnement rigoureux fondé sur une conception modulaire	R1 R2 Si les cibles de modularité sont définies, justifiées et atteintes	R1 R2 Si les cibles de modularité sont définies, justifiées et atteintes	R1 R2 Si les cibles de modularité sont définies, justifiées et atteintes	R1 Si les modules non affectés par la défaillance d'un module contribuent à la réduction / rétablissement R3 Si la tolérance aux anomalies particulières peut être prise en charge par un raisonnement rigoureux	R1 Si les modules pouvant subir l'influence d'événements externes susceptibles d'affecter simultanément plusieurs canaux, sont identifiés et soumis à une vérification approfondie R3 Si la tolérance aux anomalies particulières peut être prise en charge par un raisonnement rigoureux

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés							
		Complétude par rapport à la spécification des exigences pour le logiciel de sécurité	**Exactitude par rapport à la spécification des exigences pour le logiciel de sécurité**	**Insensibilité aux anomalies de conception intrinsèques**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies**	**Protection contre les défaillances de cause commune dues à des événements externes**
8	Utilisation de composants logiciels sécurisés/vérifiés (s'ils existent)	–	R1 R2 R3 Si l'élément contribue fortement aux exigences de sécurité particulières, et est utilisé correctement	R1 R2 R3 Réutilisation des éléments éprouvés. Cette capabilité doit être justifiée pour l'élément	R1 L'approche modulaire décompose la complexité globale en unités compréhensibles	R1 R2 R3 Réutilisation des éléments éprouvés.	–	R1 R2 Si les capacités de tolérance aux anomalies sont immédiatement fournies par l'élément et sont utilisées correctement, ou si une couche de tolérance aux anomalies entoure l'élément	R1 R2 Si des moyens de protection contre les événements externes susceptibles d'affecter simultanément plusieurs canaux, sont fournis immédiatement par l'élément et sont correctement utilisés, ou si une couche de protection entoure l'élément
9	Traçabilité ascendante entre la spécification des exigences pour la sécurité du logiciel et l'architecture du logiciel	R1 Confiance dans le fait que l'architecture traite les exigences pour la sécurité du logiciel	–	–	–	–	–	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés							
		Complétude par rapport à la spécification des exigences pour le logiciel de sécurité	**Exactitude par rapport à la spécification des exigences pour le logiciel de sécurité**	**Insensibilité aux anomalies de conception intrinsèques**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies**	**Protection contre les défaillances de cause commune dues à des événements externes**
10	Traçabilité descendante entre l'architecture du logiciel et la spécification des exigences pour la sécurité du logiciel	–	R1 Confiance dans le fait que l'architecture ne comporte pas de complexité inutile	–	–	–	–	–	–
11a	Méthodes diagrammatiques structurées	–	R1	–	R1 (Meilleure compréhension des descriptions graphiques)	–	R1 (Plus grande facilité de vérification et d'essai des conceptions structurées)	–	–
11b	Méthodes semi-formelles	R1 Méthode et notation de spécification conviviale pour l'application ou spécifique au domaine	R1 Méthode et notation de spécification conviviale pour l'application ou spécifique au domaine	R2 Peut détecter les incohérences internes, tout comportement défaillant ou les expressions mathématiquement incohérentes	–	R2 (Fournit des preuves de prévisibilité)	R2 (Fournit des preuves pour la cohérence interne du modèle de conception)	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés							
		Complétude par rapport à la spécification des exigences pour le logiciel de sécurité	**Exactitude par rapport à la spécification des exigences pour le logiciel de sécurité**	**Insensibilité aux anomalies de conception intrinsèques**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies**	**Protection contre les défaillances de cause commune dues à des événements externes**
11c	Méthodes formelles de conception et d'affinement	R1 Méthode et notation de spécification conviviale pour l'application ou spécifique au domaine	R1 Fournit une définition précise des aspects limités du comportement devant être approprié au domaine	R3 Peut détecter les incohérences internes, tout comportement défaillant ou les expressions mathématiquement incohérentes	– Note: Peut rendre plus complexe l'obtention de cette propriété.	R2 Fournit une preuve de prévisibilité	R2	–	–
11d	Génération logicielle automatique	R1 Si le logiciel exécutable est généré automatiquement du fait de la spécification des exigences, ou à partir d'une conception qui s'est révélée achevée R2 S'il s'avère que les outils de génération ont une certification appropriée	R1 Si le logiciel exécutable est généré automatiquement du fait de la spécification des exigences, ou à partir d'une conception qui s'est révélée correcte R2 S'il s'avère que les outils de génération ont une certification appropriée	R1 Si les outils de génération garantissent l'évitement d'anomalies de conception intrinsèques particulières R2 S'il s'avère que les outils de génération ont une certification appropriée	–	–	–	R1 R2 R3 Si les capacités de tolérance aux anomalies sont générées automatiquement	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés							
		Complétude par rapport à la spécification des exigences pour le logiciel de sécurité	**Exactitude par rapport à la spécification des exigences pour le logiciel de sécurité**	**Insensibilité aux anomalies de conception intrinsèques**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies**	**Protection contre les défaillances de cause commune dues à des événements externes**
12	Outils de spécification et de conception assistés par ordinateur	R1 Encapsulation des connaissances de domaine de l'EUC et de l'environnement du logiciel R2 Si la liste de contrôle des problèmes à prendre en considération est définie, justifiée et traitée	R1 Application de la traçabilité des exigences relatives au caractère descendant Techniques de simulation fonctionnelle R2 Simulation fonctionnelle selon des critères de couverture définis et justifiés	R2 Vérifications sémantique et syntaxique afin de s'assurer que les règles correspondantes sont satisfaites	R1 Animation et exploration	–	R2 Vérifications sémantique et syntaxique afin de s'assurer que les règles correspondantes sont satisfaites	–	–
13a	Comportement cyclique, avec temps de cycle maximum garanti	–	R1 pour les aspects de synchronisation de la spécification R3 Si le temps de cycle maximum est établi par un raisonnement rigoureux	R1 pour les aspects de synchronisation de la spécification R3 Si le temps de cycle maximum est établi par un raisonnement rigoureux	–	R1 pour les aspects de synchronisation de la spécification R3 Si le temps de cycle maximum est établi par un raisonnement rigoureux	R1 pour les aspects de synchronisation de la spécification R3 Si le temps de cycle maximum est établi par un raisonnement rigoureux	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés							
		Complétude par rapport à la spécification des exigences pour le logiciel de sécurité	**Exactitude par rapport à la spécification des exigences pour le logiciel de sécurité**	**Insensibilité aux anomalies de conception intrinsèques**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies**	**Protection contre les défaillances de cause commune dues à des événements externes**
13b	Architecture à déclenchement temporel	R3 Garantie de la complétude par allocation (uniquement pour les propriétés de synchronisation)	R3 Garantie de la complétude par allocation (uniquement pour les propriétés de synchronisation	R3 Garantie rigoureuse contre les anomalies de synchronisation intrinsèques	R1 Notation définie réduisant fortement toute mauvaise compréhension, prévisibilité perçue comme une approche	R3 Perturbations préjudiciables: séparation totale dans le domaine temporel, aucune perturbation	R3 Réduit fortement les efforts nécessaires pour l'essai et la certification du système	R2 Implémentation transparente de la tolérance aux anomalies	R3 Les interruptions extérieures ne peuvent pas altérer le programme à déclenchement temporel qui accorde la priorité aux tâches critiques pour la sécurité
13c	Dirigé par les événements, avec temps de réponse maximal garanti	–	–	–	R1 Les architectures dirigées par les événements peuvent empêcher l'intelligibilité	R2 Les architectures dirigées par les événements peuvent empêcher l'intelligibilité	R1 Rend les essais davantage prévisibles	–	–
14	Allocation de ressources statiques	R1	R1	R1	R1 Rend la conception plus compréhensible	R2 Avec utilisation des ressources qui définissent l'architecture	R1 Rend les essais davantage prévisibles	–	–
15	Synchronisation statique de l'accès aux ressources partagées	–	R1 Donne la prévisibilité d'accès aux ressources	R1 R3 si pris en charge par un raisonnement rigoureux par rapport à l'exactitude de synchronisation	R1 Rend la conception plus compréhensible	R1 R3 si pris en charge par un raisonnement rigoureux par rapport à l'exactitude de synchronisation	–	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Tableau C.3 – Propriétés relatives à l'intégrité systématique - Conception et développement du logiciel – outils de support et langage de programmation

(Voir 7.4.4. Référencés dans le Tableau A.3)

Technique/Mesure		Propriétés		
		Aide à la production du logiciel avec les propriétés du logiciel requises	**Clarté de l'exploitation et de la fonctionnalité de l'outil**	**Exactitude et répétabilité de la sortie**
1	Langage de programmation approprié	R2 si typage fort, conversion de type limitée. R3 si sémantique définie pour un raisonnement rigoureux	–	–
2	Langage de programmation fortement typé	R2	–	–
3	Sous-ensemble de langage	R2 Selon le sous-ensemble choisi	R1	R2 Selon le sous-ensemble choisi
4a	Outils certifiés	–	R2	R2
4b	Outils: confiance accrue résultant de l'utilisation	R1 Si la classe des erreurs de programme détectées est définie de manière systématique R2 S'il existe des preuves de validation objectives du fonctionnement des outils.	R1 Si le support de l'outil n'est pas spécifique au domaine du problème. R2 Si le support de l'outil est véritablement spécialisé dans le domaine du problème.	R1 R2 S'il existe des preuves de validation objectives du fonctionnement des outils, par exemple, une suite de validation satisfaisante

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Tableau C.4 – Propriétés relatives à l'intégrité systématique – Conception et développement du logiciel – conception détaillée (comprend la conception du système logiciel, la conception des modules logiciels et le codage)

(Voir 7.4.5 et 7.4.6. Référencés dans le Tableau A.4)

Technique/Mesure		Propriétés							
		Complétude par rapport à la spécification des exigences pour la sécurité du logiciel	**Exactitude par rapport à la spécification des exigences pour la sécurité du logiciel**	**Insensibilité aux anomalies de conception intrinsèques**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies / Détection d'anomalies**	**Insensibilité aux défaillances de cause commune**
1a	Méthodes structurées	R2	R1	R1	–	–	R1 Les conceptions structurées sont plus facilement vérifiables et peuvent être soumises à essai	–	–
1b	Méthodes semi-formelles	R2	R2	R2	–	R2	R2	–	–
1c	Méthodes formelles de conception et d'affinement	–	R3	R3	– Note: Peut rendre plus complexe l'obtention de cette propriété.	R3 Fournit des preuves de prévisibilité	R2	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés							
		Complétude par rapport à la spécification des exigences pour la sécurité du logiciel	**Exactitude par rapport à la spécification des exigences pour la sécurité du logiciel**	**Insensibilité aux anomalies de conception intrinsèques**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies / Détection d'anomalies**	**Insensibilité aux défaillances de cause commune**
2	Outils de conception assistés par ordinateur	R2 Dépend de l'outil de spécification assisté par ordinateur qui applique les vérifications sémantique et syntaxique afin de s'assurer que les règles correspondantes sont satisfaites	R1	R2 Dépend de l'outil de spécification assisté par ordinateur qui applique les vérifications sémantique et syntaxique afin de s'assurer que les règles correspondantes sont satisfaites	–	–	R2 Dépend de l'outil CASE venant à l'appui de la couverture d'essai et de la vérification statique	–	–
3	Programmation défensive	–	–	–	– Note: Peut rendre plus complexe l'obtention de cette propriété.	– Note: Peut rendre plus complexe l'obtention de cette propriété.	–	R1 (R2 si des cibles de couverture sont définies, justifiées et atteintes)	R1 (R2 si des cibles de couverture sont définies, justifiées et atteintes)
4	Approche modulaire	–	–	R1	R1	R1	R1	–	–
5	Règles de conception et de codage	–	–	R1	R1	R1	R1	–	–
6	Programmation structurée	–	R1	R1	R1	R1	R1	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

	Technique/Mesure	Propriétés							
		Complétude par rapport à la spécification des exigences pour la sécurité du logiciel	**Exactitude par rapport à la spécification des exigences pour la sécurité du logiciel**	**Insensibilité aux anomalies de conception intrinsèques**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies / Détection d'anomalies**	**Insensibilité aux défaillances de cause commune**
7	Utilisation de composants logiciels sécurisés/vérifiés (s'ils existent)	–	–	R1 Réutilisation des éléments éprouvés	R1 L'approche modulaire décompose la complexité globale en unités compréhensibles	R1 Le comportement de l'élément est déjà connu	–	–	–
8	Traçabilité ascendante entre la spécification des exigences pour la sécurité du logiciel et la conception du logiciel	R1 Confiance dans le fait que la conception traite les exigences pour la sécurité du logiciel	–	–	–	–	–	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Tableau C.5 – Propriétés relatives à l'intégrité systématique – Conception et développement du logiciel – essai et intégration des modules logiciels

(Voir 7.4.7 et 7.4.8. Référencés dans le Tableau A.5)

Technique/Mesure		Propriétés			
		Complétude de l'essai et de l'intégration par rapport à la spécification de conception du logiciel	**Exactitude de l'essai et de l'intégration par rapport à la spécification de conception du logiciel (réalisation satisfaisante)**	**Répétabilité**	**Configuration d'essai définie avec précision**
1	Essai probabiliste	R1 (R2 si des cibles de couverture de profil opérationnel sont définies, justifiées et atteintes)	R1 (R2 si les sorties requises sont définies, justifiées et atteintes)	–	–
2	Analyse dynamique et essai	R1 (R2 si des cibles de couverture structurelle sont définies, justifiées et atteintes)	R1 (R2 si les sorties requises sont définies, justifiées et atteintes)	–	–
3	Enregistrement et analyse des données	–	R1	R1 Favorise la cohérence des procédures d'essai	R2 Si les registres d'anomalies/journaux d'essai comprennent les détails du référentiel du logiciel
4	Essais fonctionnels et boîte noire	R1 (R2 si des cibles de couverture de profil opérationnel sont définies, justifiées et atteintes)	R1 (R2 si les sorties requises sont définies, justifiées et atteintes)	–	–
5	Essais de fonctionnement	–	R1 (R2 si les sorties requises sont définies, justifiées et atteintes)	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés			
		Complétude de l'essai et de l'intégration par rapport à la spécification de conception du logiciel	**Exactitude de l'essai et de l'intégration par rapport à la spécification de conception du logiciel (réalisation satisfaisante)**	**Répétabilité**	**Configuration d'essai définie avec précision**
6	Essais basés sur le modèle (MBT)	R2 Les MBT présentent au stade initial les ambiguïtés de la spécification et de la conception; le processus MBT commence par des exigences R3 Si application d'un raisonnement rigoureux à la modélisation et de la génération de cas d'essai	R2 L'évaluation des résultats et séquences d'essai de régression représente un avantage essentiel des MBT R3 L'application d'une approche de modélisation rigoureuse rend possible la fourniture de preuves de couverture objectives	R3 Les MBT (et la TCG) ont pour objectif l'exécution automatique d'essais générés	R2 Les MBT sont automatisés et la configuration d'essai doit être définie avec précision; l'exécution des essais générés est similaire aux essais boîte noire, avec possibilité d'une combinaison avec la mesure de la couverture du niveau de code source
7	Essais d'interface	–	R1 (R2 si les sorties requises sont définies, justifiées et atteintes)	–	–
8	Outils de gestion et d'automatisation des essais	R1 (R2 si des cibles de couverture d'essai sont définies, justifiées et atteintes)	–	R1 L'automatisation favorise la cohérence	R2 Produit la répétabilité des essais
9	Traçabilité ascendante entre la spécification des exigences pour la sécurité du logiciel et les spécifications des modules et de l'essai d'intégration	R1 Confiance dans le fait que la conception d'essai traite les exigences pour la sécurité du logiciel	–	–	R2 Confiance dans un référentiel clair des exigences en essai
10	Vérification formelle	R3 Si un raisonnement rigoureux est appliqué à la construction de cas d'essai visant à montrer que tous les aspects de la conception ont été pris en compte	R3 Fournit des preuves objectives de la satisfaction à toutes les exigences pour la sécurité du logiciel	R1 Si indisponibilité des outils de support R2 Si outil pris en charge	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Tableau C.6 – Propriétés relatives à l'intégrité systématique – Intégration de l'électronique programmable (matériel et logiciel)

(Voir 7.5. Référencés dans le Tableau A.6)

	Technique/Mesure	Propriétés			
		Complétude de l'intégration par rapport aux spécifications de conception	**Exactitude de l'intégration par rapport aux spécifications de conception (réalisation satisfaisante)**	**Répétabilité**	**Configuration d'intégration définie avec précision**
1	Essais fonctionnels et boîte noire	R1 (R2 si des cibles de couverture de profil opérationnel sont définies, justifiées et atteintes)	R1 (R2 si les sorties requises sont définies, justifiées et atteintes)	–	–
2	Essais de fonctionnement	–	R1 (R2 si les sorties requises sont définies, justifiées et atteintes)	–	–
3	Traçabilité ascendante entre le système et les exigences de conception du logiciel pour l'intégration du matériel/logiciel et les spécifications d'essai d'intégration du matériel/logiciel	R1 Confiance dans le fait que les spécifications d'essai d'intégration du matériel/logiciel traitent les exigences d'intégration	–	–	R2 Confiance dans un référentiel clair des exigences en essai

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Tableau C.7 – Propriétés relatives à l'intégrité systématique – Validation de sécurité du logiciel

(Voir 7.7. Référencées dans le Tableau A.7)

Technique/Mesure		Propriétés			
		Complétude de la validation par rapport à la spécification de conception du logiciel	**Exactitude de la validation par rapport à la spécification de conception du logiciel (réalisation satisfaisante)**	**Répétabilité**	**Configuration de validation définie avec précision**
1	Essai probabiliste	R1 (R2 si des cibles de couverture de profil opérationnel sont définies, justifiées et atteintes)	R1 (R2 si les sorties requises sont définies, justifiées et atteintes)	–	–
2	Simulation de processus	R1	R1 (R2 si les sorties requises sont définies, justifiées et atteintes)	–	R2 Donne une définition de l'environnement externe
3	Essais fonctionnels et boîte noire	R1 (R2 si des cibles de couverture de profil opérationnel sont définies, justifiées et atteintes)	R1 (R2 si les sorties requises sont définies, justifiées et atteintes)	–	–
4	Traçabilité ascendante entre la spécification des exigences pour la sécurité du logiciel et la planification de la validation de sécurité du logiciel	R1 Confiance dans le fait que la planification de la validation de sécurité du logiciel traite les exigences pour la sécurité du logiciel	–	–	R2 Confiance dans un référentiel clair des exigences en essai
5	Traçabilité descendante entre la planification de la validation de sécurité du logiciel et la spécification des exigences pour la sécurité du logiciel	–	R1 Confiance dans le fait que la planification de la validation de sécurité du logiciel ne comporte pas de complexité inutile	–	R2 Confiance dans un référentiel clair des exigences en essai

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Tableau C.8 – Propriétés relatives à l'intégrité systématique – Modification du logiciel

(Voir 7.8. Référencées dans le Tableau A.8)

Technique/Mesure		Propriétés					
		Complétude de la modification par rapport à ses exigences	**Exactitude de la modification par rapport à ses exigences**	**Insensibilité à l'introduction d'anomalies de conception intrinsèques**	**Evitement de comportement indésirable**	**Conception vérifiable et pouvant être soumise à essai**	**Essais de régression et couverture de vérification**
1	Analyse d'impact	–	–	–	R1	R1	R1
2	Revérification du module logiciel modifié	R1 (R2 si existence de cibles de vérification objectives)	R1 (R2 si existence de cibles de vérification objectives)	R1 (R2 si existence de cibles de vérification objectives)	–	–	R1/R2R1 (R2 si existence de cibles de vérification objectives)
3	Revérification des modules logiciels affectés	R1 (R2 si existence de cibles de vérification objectives)	R1 (R2 si existence de cibles de vérification objectives)	R1 (R2 si existence de cibles de vérification objectives)	–	–	R1 (R2 si existence de cibles de vérification objectives)
4a	Revalidation du système complet	R1 (R2 si existence de cibles de vérification objectives)	R1 (R2 si existence de cibles de vérification objectives)	–	R1 (R2 si existence de cibles de vérification objectives)	–	R1 (R2 si existence de cibles de vérification objectives)
4b	Validation de non-régression	R1 (R2 si existence de cibles de vérification objectives)	R1 (R2 si existence de cibles de vérification objectives)	–	R1 (R2 si existence de cibles de vérification objectives)	–	R1 (R2 si existence de cibles de vérification objectives)
5	Gestion de configuration logicielle	–	–	–	–	–	R1
6	Enregistrement et analyse des données	R1	R1	–	–	–	–

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés					
		Complétude de la modification par rapport à ses exigences	**Exactitude de la modification par rapport à ses exigences**	**Insensibilité à l'introduction d'anomalies de conception intrinsèques**	**Evitement de comportement indésirable**	**Conception vérifiable et pouvant être soumise à essai**	**Essais de régression et couverture de vérification**
7	Traçabilité ascendante entre la spécification des exigences pour la sécurité du logiciel et le plan de modification du logiciel (y compris la revérification et la revalidation)	R1 Confiance dans le fait que le plan de modification du logiciel (y compris la revérification et la revalidation) traite les exigences pour la sécurité du logiciel	–	–	–	–	–
8	Traçabilité descendante entre le plan de modification du logiciel (y compris la revérification et la revalidation) et la spécification des exigences pour la sécurité du logiciel	–	R1 Confiance dans le fait que le plan de modification du logiciel (y compris la revérification et la revalidation) ne comporte pas de complexité inutile	–	–	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Tableau C.9 – Propriétés relatives à l'intégrité systématique – Vérification du logiciel

(Voir 7.9. Référencées dans le Tableau A.9)

Technique/Mesure		Propriétés			
		Complétude de la vérification par rapport à la phase précédente	**Exactitude de la vérification par rapport à la phase précédente (réalisation satisfaisante)**	**Répétabilité**	**Configuration de vérification définie avec précision**
1	Preuve formelle	–	R3	–	–
2	Animation de la spécification et de la conception	R1	R1	–	–
3	Analyse statique	–	R1/R2/R3 (La rigueur peut aller de l'application d'un sous-ensemble de langages à une analyse formelle mathématique)	–	–
4	Analyse dynamique et essai	R1 (R2 si des cibles de couverture structurelle sont définies, justifiées et atteintes)	R1 (R2 si les sorties requises sont définies, justifiées et atteintes)	–	–
5	Traçabilité ascendante entre la spécification de conception du logiciel et le plan de vérification du logiciel (y compris la vérification des données).	R1 Confiance dans le fait que le plan de vérification du logiciel (y compris la vérification des données) traite les exigences pour la sécurité du logiciel	–	–	R2 Confiance dans un référentiel clair des exigences en essai
6	Traçabilité descendante entre la planification de la vérification du logiciel (y compris la vérification des données) et la spécification de conception du logiciel	–	R1 Confiance dans le fait que le plan de vérification du logiciel (y compris la vérification des données) ne comporte pas de complexité inutile	–	R2 Confiance dans un référentiel clair des exigences en essai
7	Analyse numérique hors ligne	–	R1 plus grande confiance dans la précision numérique prévue de calculs bien établis (R2 avec des critères d'acceptation objectifs). R3 si utilisation conjointe avec un raisonnement systématique objectif visant à justifier les critères d'acceptation)	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Tableau C.10 – Propriétés relatives à l'intégrité systématique – Évaluation de la sécurité fonctionnelle

(Voir Article 8. Référencées dans le Tableau A.10)

Technique/Mesure		Propriétés						
		Complétude de l'évaluation de la sécurité fonctionnelle par rapport à la présente norme	**Exactitude de l'évaluation de la sécurité fonctionnelle par rapport aux spécifications de conception (réalisation satisfaisante)**	**Résolution traçable de tous les problèmes identifiés**	**Aptitude à modifier l'évaluation de la sécurité fonctionnelle après modification sans la nécessité d'un remaniement étendu de l'évaluation**	**Répétabilité**	**Rapidité d'exécution**	**Configuration définie avec précision**
1	Listes de contrôle	R1	R1	R1	–	R1	–	–
2	Tables de décision/de vérité	R1	R2	–	–	R2	–	–
3	Analyse des défaillances	R2	R2	R1 (L'analyse des défaillances est basée sur des listes de défaillances convenues)	–	R1 (L'analyse des défaillances est basée sur des listes de défaillances convenues)	–	–
4	Analyse des défaillances de cause commune d'un logiciel diversifié (si un logiciel diversifié est effectivement utilisé)	R2	R2	R1 (sous réserve que l'analyse CCF soit fondée sur des listes d'initiateur CC)	–	R1 (sous réserve que l'analyse CCF soit fondée sur des listes d'initiateur CC)	–	–
5	Diagramme de blocs de fiabilité	R1	R1	–	–	–	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés						
		Complétude de l'évaluation de la sécurité fonctionnelle par rapport à la présente norme	**Exactitude de l'évaluation de la sécurité fonctionnelle par rapport aux spécifications de conception (réalisation satisfaisante)**	**Résolution traçable de tous les problèmes identifiés**	**Aptitude à modifier l'évaluation de la sécurité fonctionnelle après modification sans la nécessité d'un remaniement étendu de l'évaluation**	**Répétabilité**	**Rapidité d'exécution**	**Configuration définie avec précision**
6	Traçabilité ascendante entre les exigences de l'Article 8 de la CEI 61508-3 et le plan pour l'évaluation de la sécurité fonctionnelle du logiciel	R1 Confiance dans le fait que le plan pour l'évaluation de la sécurité fonctionnelle du logiciel traite les exigences de l'Article 8 de la CEI 61508-3	–	–	–	–	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

C.3 Propriétés relatives à l'intégrité systématique – Tableaux détaillés

Tableau C.11 – Propriétés détaillées – Conception et règles de codage

(Référencées dans le Tableau B.1)

Technique/Mesure		Propriétés							
		Complétude par rapport à la spécification des exigences pour la sécurité du logiciel	**Exactitude par rapport à la spécification des exigences pour la sécurité du logiciel**	**Insensibilité aux anomalies de conception intrinsèques**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies / Détection d'anomalies**	**Insensibilité aux défaillances de cause commune**
1	Utilisation de règles de codage pour réduire la probabilité d'erreurs	–	–	R1	R1 Elimine les éléments de langage sélectionnés	R1	R1	–	–
2	Pas d'objets dynamiques	–	–	R1/R2/R3 Selon le langage utilisé	–	R1/R2/R3 Selon le langage utilisé	R1/R2 Selon le langage utilisé	–	–
3a	Pas de variables dynamiques	–	–	R1/R2/R3 Selon le langage utilisé	–	R1/R2/R3 Selon le langage utilisé	R1/R2 Selon le langage utilisé	–	–
3b	Contrôle en ligne pendant la création de variables dynamiques	–	–	R1/R2/R3 Selon le langage utilisé	–	R1/R2/R3 Selon le langage utilisé	R1/R2 Selon le langage utilisé	–	–
4	Utilisation limitée des interruptions	–	–	R1/R2 Selon le langage utilisé	R1 Accroît la clarté de la logique et des séquences d'événements	R1/R2 Selon le langage utilisé	R1/R2 Selon le langage utilisé	–	–
5	Utilisation limitée des pointeurs	–	–	R1/R2 Selon le langage utilisé	R1 Accroît la clarté de la logique	R1/R2 Selon le langage utilisé	R1/R2 Selon le langage utilisé	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés							
		Complétude par rapport à la spécification des exigences pour la sécurité du logiciel	**Exactitude par rapport à la spécification des exigences pour la sécurité du logiciel**	**Insensibilité aux anomalies de conception intrinsèques**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies / Détection d'anomalies**	**Insensibilité aux défaillances de cause commune**
6	Utilisation limitée de la récursion	–	–	R1/R2 Selon le langage utilisé	–	R1/R2 Selon le langage utilisé	R1/R2 Selon le langage utilisé	–	–
7	Pas de branchements inconditionnels dans les programmes en langages de haut niveau	–	–	R1/R2 Selon le langage utilisé	R1 Accroît la clarté de la logique	R1/R2 Selon le langage utilisé	R1/R2 Selon le langage utilisé	–	–
8	Pas de conversion de type automatique	–	R2 Evite les erreurs d'arrondi	R2 Evite les erreurs d'arrondi	R1	R1	–	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

---,,,,,,,,,,-,,,,,,,,,,,.....,,,,,,--

Tableau C.12 – Propriétés détaillées – Analyse dynamique et essais

(Référencées dans le Tableau B.2)

Technique/Mesure		Propriétés			
		Complétude de l'essai et vérification par rapport aux spécifications de conception du logiciel	**Exactitude de l'essai et vérification par rapport aux spécifications de conception du logiciel (réalisation satisfaisante)**	**Répétabilité**	**Configuration d'essai et de vérification définie avec précision**
1	Exécution de cas d'essai à partir de l'analyse des valeurs aux limites	–	R1 (R2 si existence de critères objectifs pour résultats aux limites)	–	–
2	Exécution de cas d'essai à partir de l'estimation des erreurs	–	R1	–	–
3	Exécution de cas d'essai à partir de l'implantation d'erreurs	–	R1	–	–
4	Exécution de cas d'essai à partir de la génération de cas d'essai basés sur les modèles	R2 Le processus MBT commence par des exigences et facilite la détermination précoce des erreurs lors de la conception et du développement du logiciel R3 si un raisonnement rigoureux est appliqué à la modélisation, et si la TCG (génération de cas d'essai) est appliquée	R2 L'évaluation des résultats et séquences d'essai de régression représente un avantage essentiel des MBT; elle facilite par ailleurs la compréhension des conséquences des exigences spécifiées R3 L'application d'une approche de modélisation rigoureuse rend possible la fourniture de preuves de couverture objectives	R3 Les MBT (et la TCG) ont pour objectif l'exécution automatique d'essais générés	R2 Les MBT sont automatisés et la configuration d'essai doit être définie avec précision; l'exécution des essais générés est similaire aux essais boîte noire, avec possibilité d'une combinaison avec la mesure de la couverture du niveau de code source
5	Modélisation du fonctionnement	–	R1 (R2 si application d'exigences objectives relatives au fonctionnement)	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés			
		Complétude de l'essai et vérification par rapport aux spécifications de conception du logiciel	**Exactitude de l'essai et vérification par rapport aux spécifications de conception du logiciel (réalisation satisfaisante)**	**Répétabilité**	**Configuration d'essai et de vérification définie avec précision**
6	Classes d'équivalence et essai des partitions d'entrée	R1 (si le profil de données d'entrée est parfaitement défini et si sa structure est simple d'un point de vue pratique)	R1 (si les partitions ne contiennent vraisemblablement aucune non linéarité, c'est-à-dire si tous les membres d'une classe sont véritablement équivalents)	–	–
7	Essais basés sur la structure	–	R1 (R2 si existence de cibles de couverture structurelles objectives)	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Tableau C.13 – Propriétés détaillées – Essais fonctionnels et boîte noire

(Références dans le Tableau B.3)

Technique/Mesure		Propriétés			
		Complétude de l'essai, de l'intégration et de la validation par rapport aux spécifications de conception	**Exactitude des essais, de l'intégration et de la validation par rapport aux spécifications de conception (réalisation satisfaisante)**	**Répétabilité**	**Configuration d'essai, d'intégration et de validation définie avec précision**
1	Exécution de cas d'essai à partir de diagrammes cause/conséquence	R1	R1	–	–
2	Exécution de cas d'essai à partir de la génération de cas d'essai basés sur les modèles	R2 Les MBT (essais basés sur les modèles) consistent en la génération automatique de cas/procédures d'essai efficaces qui utilisent des modèles d'exigences système et de fonctionnalité spécifiée; ils facilitent la détermination précoce des erreurs et la compréhension des conséquences des exigences spécifiées R3 Si un raisonnement rigoureux est appliqué, et si la TCG est utilisée	R2 Les MBT sont basés sur des modèles de systèmes issus d'exigences (principalement fonctionnelles/liées au comportement). R3 L'application d'une approche de modélisation rigoureuse rend possible l'existence de preuves de couverture objectives	R3 Les MBT (et la TCG) ont pour objectif l'exécution automatique d'essais générés	R2 Les MBT sont automatisés et la configuration d'essai doit être définie avec précision
3	Prototypage/animation	–	R1	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

	Technique/Mesure	Propriétés			
		Complétude de l'essai, de l'intégration et de la validation par rapport aux spécifications de conception	**Exactitude des essais, de l'intégration et de la validation par rapport aux spécifications de conception (réalisation satisfaisante)**	**Répétabilité**	**Configuration d'essai, d'intégration et de validation définie avec précision**
4	Classes d'équivalence et essai des partitions d'entrée, y compris l'analyse des valeurs aux limites	R1 (si le profil de données d'entrée est parfaitement défini et si sa structure est simple d'un point de vue pratique)	R1 (si les partitions ne contiennent vraisemblablement aucune non linéarité, c'est-à-dire si tous les membres d'une classe sont véritablement équivalents)	–	–
5	Simulation de processus	–	R1	–	R2 Donne une définition de l'environnement externe

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Tableau C.14 – Propriétés détaillées – Analyse des défaillances

(Références dans le Tableau B.4)

Technique/Mesure		Propriétés						
		Complétude de l'évaluation de la sécurité fonctionnelle par rapport à la présente norme	**Exactitude de l'évaluation de la sécurité fonctionnelle par rapport aux spécifications de conception (réalisation satisfaisante)**	**Résolution traçable de tous les problèmes identifiés**	**Aptitude à modifier l'évaluation de la sécurité fonctionnelle après modification sans la nécessité d'un remaniement étendu de l'évaluation**	**Répétabilité**	**Rapidité d'exécution**	**Configuration définie avec précision**
1a	Diagrammes cause/conséquence	R2	R2	–	–	–	–	–
1b	Analyse par arbre d'événement	R2	R2	–	–	–	–	–
2	Analyse par arbre de panne	R2	R2	–	–	–	–	–
3	Analyse des défaillances fonctionnelles du logiciel	R2	R2	–	–	–	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Tableau C.15 – Propriétés détaillées – Modélisation

(Références dans le Tableau B.5)

Technique/Mesure		Propriétés			
		Complétude de la validation par rapport à la spécification de conception du logiciel	Exactitude de la validation par rapport à la spécification de conception du logiciel (réalisation satisfaisante)	Répétabilité	Configuration de validation définie avec précision
1	Diagrammes de flux de données	_	R1	_	_
2a	Automate à états finis	R3	R3	_	_
2b	Méthodes formelles	R3	R3	_	_
2c	Réseaux de Pétri temporels	_	R1	_	_
3	Modélisation du fonctionnement	_	R1	_	_
4	Prototypage/animation	_	R1	_	_
5	Diagrammes de structures	_	R1	_	_

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Tableau C.16 – Propriétés détaillées – Essais de fonctionnement

(Références dans le Tableau B.6)

Technique/Mesure		Propriétés			
		Complétude de l'essai et de l'intégration par rapport aux spécifications de conception	**Exactitude de l'essai et de l'intégration par rapport aux spécifications de conception (réalisation satisfaisante)**	**Répétabilité**	**Configuration d'essai et d'intégration définie avec précision**
1	Essais d'avalanche/de stress	–	R1 (R2 si des cibles objectives sont fixées)	–	–
2	Temps de réponse et contraintes mémoire	–	R1 (R2 si des cibles objectives sont fixées)	–	–
3	Exigences relatives au fonctionnement	–	R1 (R2 si des cibles objectives sont fixées)	–	–

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Tableau C.17 – Propriétés détaillées – Méthodes semi-formelles

(Référencées dans le Tableau B.7)

Technique/Mesure		Propriétés									
		Complétude par rapport à la spécification des exigences pour la sécurité du logiciel	**Exactitude par rapport à la spécification des exigences pour la sécurité du logiciel**	**Insensibilité aux anomalies de conception intrinsèques**	**Intelligibilité des exigences de sécurité**	**Insensibilité aux perturbations préjudiciables des fonctions non liées à la sécurité avec les besoins de sécurité devant être traités par le logiciel**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies / Détection d'anomalies**	**Insensibilité aux défaillances de cause commune dues à des événements externes**
1	Diagrammes de blocs logiques / fonctionnels	R2	R2	R2	–	–	R1	R2	–	–	R1
2	Diagrammes de séquence	R2	R2	R2	–	–	R1	R2	–	–	R2
3	Diagrammes de flux de données	R1	R1	R1	–	–	R1 Convient au traitement de transactions	–	–	–	R1
4a	Automates finis / diagrammes de changement d'états	R2	R2	R2	–	–	R1 Spécification mathématique complète des séquences d'événements	R2	–	–	R2
4b	Réseaux de Pétri temporels	R2	R2	R2	–	–	R1 Spécifie des interactions en temps réel	R2	–	–	R2
5	Modèles de données entité-relation-attribut	R1	R1	R1	–	–	R1	–	–	–	R1
6	Diagrammes de séquences de messages	R2	R2	R2	–	–	R1	R2	–	–	R2

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés									
		Complétude par rapport à la spécification des exigences pour la sécurité du logiciel	**Exactitude par rapport à la spécification des exigences pour la sécurité du logiciel**	**Insensibilité aux anomalies de conception intrinsèques**	**Intelligibilité des exigences de sécurité**	**Insensibilité aux perturbations préjudiciables des fonctions non liées à la sécurité avec les besoins de sécurité devant être traités par le logiciel**	**Simplicité et intelligibilité**	**Prévisibilité du comportement**	**Conception vérifiable et pouvant être soumise à essai**	**Tolérance aux anomalies / Détection d'anomalies**	**Insensibilité aux défaillances de cause commune dues à des événements externes**
7	Tables de décision/de vérité	R2	R2	R2	–	–	R1 Pour une logique combinatoire	R2	–	–	R2

Tableau C.18 – Propriétés relatives à l'intégrité systématique – Analyse statique

(Références dans le Tableau B.8)

Technique/Mesure		Propriétés			
		Complétude de la vérification par rapport à la phase précédente	**Exactitude de la vérification par rapport à la phase précédente (réalisation satisfaisante)**	**Répétabilité**	**Configuration de vérification définie avec précision**
1	Analyse des valeurs aux limites	–	R1 (R2 si existence de critères objectifs pour résultats aux limites)	–	–
2	Listes de contrôle	–	R1	–	R1
3	Analyse du flux de commande	–	R1	–	–
4	Analyse du flux de données	–	R1	–	–
5	Estimation des erreurs	–	R1	–	–
6a	Inspections formelles, y compris les critères spécifiques	R2	R2	–	R2
6b	Lectures croisées (logiciel)	R1	R1	–	R1

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Technique/Mesure		Propriétés			
		Complétude de la vérification par rapport à la phase précédente	**Exactitude de la vérification par rapport à la phase précédente (réalisation satisfaisante)**	**Répétabilité**	**Configuration de vérification définie avec précision**
7	Exécution symbolique	–	R2 R3 si utilisation dans les pré- et post-conditions définies de manière formelle dans le contexte et exécution par un outil appliquant un algorithme rigoureux mathématique	–	–
8	Revue de conception	R2	R1 R2 (avec des critères objectifs)	–	R2
9	Analyse statique du comportement d'erreur d'exécution	–	R1 R3 pour certaines classes d'erreur si exécution par un outil appliquant un algorithme rigoureux mathématique	–	–
10	Analyse de la durée d'exécution la plus défavorable	R1	R3	–	R2

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Tableau C.19 – Propriétés détaillées – Approche modulaire

(Référencées dans le Tableau B.9)

	Technique/Mesure	Propriétés							
		Complétude par rapport à la spécification des exigences pour la sécurité du logiciel	Exactitude par rapport à la spécification des exigences pour la sécurité du logiciel	Insensibilité aux anomalies de conception intrinsèques	Simplicité et intelligibilité	Prévisibilité du comportement	Conception vérifiable et pouvant être soumise à essai	Tolérance aux anomalies / Détection d'anomalies	Insensibilité aux défaillances de cause commune
1	Limitation de la taille des modules logiciels	–	–	R1	R1	R1	R1	–	–
2	Maîtrise de la complexité du logiciel	–	–	R1	R1	R1	R1	–	–
3	Masquage/encapsulation des informations	–	–	R1	R1	R1	R1	–	–
4	Limitation du nombre de paramètres / nombre fixe de paramètres de sous-programme	–	–	R1	R1	R1	R1	–	–
5	Un point d'entrée/un point de sortie dans les sous-programmes et les fonctions	–	–	R1	R1	R1	R1	–	–
6	Interface totalement définie	–	–	R2	R1	R1	R1	–	–

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Annexe D
(normative)

Manuel de sécurité d'article conforme – exigences supplémentaires pour les composants logiciels

D.1 Objet du manuel de sécurité

D.1.1 Lorsqu'un composant est réutilisé ou destiné à être réutilisé dans un ou plusieurs développements de système, il est nécessaire d'assurer que le composant est accompagné d'une description suffisamment précise et complète (c'est-à-dire fonctions, contraintes et preuves), pour permettre une évaluation de l'intégrité d'une fonction de sécurité spécifique qui dépend en tout ou en partie du composant. Ceci doit être mis en œuvre au moyen d'un manuel de sécurité.

D.1.2 Le manuel de sécurité peut comprendre la documentation du fournisseur du composant si elle satisfait aux exigences de l'Annexe D de la CEI 61508-2 et de la présente annexe. Dans le cas contraire, il convient de le créer en tant que partie intégrante de la conception du système relatif à la sécurité.

D.1.3 Le manuel de sécurité doit définir les attributs d'un composant qui peuvent comprendre les contraintes du matériel et/ou du logiciel que l'intégrateur doit connaître et prendre en considération pendant l'application. Il constitue notamment le vecteur permettant de transmettre à l'intégrateur les informations relatives à ses propriétés, à l'objet pour lequel le composant a été conçu, à son comportement et à ses caractéristiques.

NOTE 1 Le domaine d'application et le délai de livraison du manuel de sécurité dépend de la personne à qui il s'applique, du type d'intégrateur, de l'objet du composant et de la personne chargée de sa fourniture et de sa maintenance.

NOTE 2 La personne, le département ou l'organisation qui intègre le logiciel est appelé(e) intégrateur.

D.2 Contenu du manuel de sécurité pour un composant logiciel

D.2.1 Le manuel de sécurité doit comporter toutes les informations requises par l'Annexe D de la CEI 61508-2 et pertinentes pour le composant. Par exemple, les articles relatifs au matériel définis dans l'Annexe D de la CEI 61508-2 ne sont pas appropriés à un composant logiciel pur.

D.2.2 Le composant doit être identifié et toutes les instructions nécessaires à son utilisation doivent être mises à la disposition de l'intégrateur.

NOTE Pour le logiciel, ceci peut être démontré par une identification claire du composant et la démonstration que son contenu n'a pas été modifié.

D.2.3 Configuration du composant:

a) La configuration du composant logiciel, de l'environnement d'exécution du logiciel et du matériel, et si nécessaire la configuration du système de compilation / liaison doivent être identifiées dans le manuel de sécurité.

b) La configuration recommandée du composant logiciel doit être documentée dans le manuel de sécurité et elle doit être utilisée dans l'application de sécurité.

c) Le manuel de sécurité doit comporter toutes les hypothèses formulées et selon lesquelles dépend la justification d'utilisation du composant.

D.2.4 Le manuel de sécurité doit comporter les informations suivantes:

--`,,```,,,,````-`-`,,`,,`,`,,`---
Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

a) Compétence: Il convient de spécifier le niveau minimal de connaissance prévue de l'intégrateur du composant, c'est-à-dire la connaissance des outils d'application spécifiques.

b) Niveau de confiance accordée au composant: Détails de toute certification du composant, évaluation indépendante réalisée, niveau d'intégrité que l'intégrateur peut accorder au composant préexistant. A cet effet, il convient de considérer l'intégrité pour laquelle le composant a été conçu, les normes et règles suivies pendant le processus de conception, et toute contrainte imposée à l'intégrateur et devant être mise en œuvre à l'appui de la capabilité systématique déclarée (selon la fonctionnalité du composant, il est admis que certaines exigences ne soient satisfaites qu'à l'étape d'intégration d'un système. Dans ce cas, ces exigences doivent être identifiées pour l'avancement des tâches de l'intégrateur. Les exigences relevant des temps de réponse et des performances en constituent deux exemples).

NOTE Contrairement à la CEI 61508-2, la CEI 61508-3 n'exige pas d'inclure des modes de défaillance du logiciel ni des taux de défaillance quantitatifs dans le manuel de sécurité du composant, dans la mesure où les causes d'erreurs logicielles sont fondamentalement différentes des causes des défaillances aléatoires du matériel concernées définies dans l'Annexe D de la CEI 61508-2.

c) Instructions d'installation: Détails du, ou référence au, mode d'installation du composant préexistant dans le système intégré.

d) Raison de la version du composant: Détails indiquant si le composant préexistant a fait l'objet d'une version pour suppression d'anomalies courantes ou pour inclusion d'une fonctionnalité supplémentaire.

e) Anomalies courantes: Il convient de fournir les détails de toutes les anomalies courantes, avec une explication de l'anomalie, son mode d'occurrence et les mécanismes que l'intégrateur doit utiliser pour corriger l'anomalie si les fonctions particulières devaient être utilisées.

f) Compatibilité ascendante: Détails indiquant si le composant est compatible avec les versions antérieures du sous-système, et dans le cas contraire, des détails du processus permettant d'accéder au chemin de mise à jour à suivre.

g) Compatibilité avec d'autres systèmes: Un composant préexistant peut dépendre d'un système d'exploitation spécialisé. Dans ce cas, il convient de fournir des informations détaillées sur la version du système d'exploitation spécialisé.

Il convient de spécifier également la norme de construction comprenant l'identification et la version du compilateur, les outils utilisés pour la création du composant préexistant (identification et version), et le composant préexistant d'essai utilisé (de nouveau identification et version).

h) Configuration du composant: Il convient de fournir des détails du (des) nom(s) et description(s) du composant préexistant, y compris la version / le numéro / l'état de la modification.

i) Gestion des modifications: Le mécanisme permettant à l'intégrateur d'introduire une demande de modification auprès du producteur du logiciel.

j) Exigences non satisfaites: Il est admis que des exigences spécifiques aient été spécifiées mais n'aient pas été satisfaites dans la révision en vigueur du composant. Dans ce cas, il convient d'identifier ces exigences pour que l'intégrateur puisse en tenir compte.

k) Etat de sécurité de conception: Dans certains cas de défaillance contrôlée de l'application du système, le composant peut revenir à un état de sécurité de conception. Dans ce cas, il convient de spécifier la définition précise de l'état de sécurité de conception pour que l'intégrateur puisse en tenir compte.

l) Contraintes d'interface: Détails de toute contrainte spécifique, notamment les exigences d'interface utilisateur doivent être identifiées.

m) Les détails des mesures de sécurité éventuelles pouvant avoir été mises en œuvre contre les menaces et les vulnérabilités répertoriées

n) Composants configurables: Les détails de la ou des méthodes de configuration disponibles pour le composant, leur utilisation et toutes contraintes éventuelles imposées à leur application doivent être fournis.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

D.3 Justification des déclarations du manuel de sécurité du composant

D.3.1 Toutes les déclarations fournies dans le manuel de sécurité du composant doivent être justifiées par des preuves documentaires appropriées. Voir 7.4.9.7 de la CEI 61508-2.

NOTE 1 Il est essentiel que la performance de sécurité déclarée d'un composant soit soutenue par des preuves suffisantes. Les déclarations non soutenues par des preuves ne permettent pas d'établir le caractère approprié et l'intégrité de la fonction de sécurité à laquelle l'élément contribue.

NOTE 2 La preuve documentaire peut être déduite de la propre documentation du fournisseur du composant et des enregistrements du processus de développement du fournisseur du composant. Elle peut également être créée ou complétée par des activités de qualification supplémentaires entreprises par le développeur du système relatif à la sécurité ou par des tierces parties.

NOTE 3 La disponibilité des preuves peut faire l'objet de restrictions d'ordre commercial ou juridique (par exemple, droit d'auteur ou droits de propriété intellectuelle). Ces restrictions ne relèvent pas du domaine d'application de la présente norme. Lorsque les preuves ne peuvent être fournies, alors l'élément ne convient pas à une utilisation avec les systèmes E/E/PE relatifs à la sécurité.

D.3.2 La preuve documentaire qui justifie les déclarations dans le manuel de sécurité du composant est différente de celle du manuel de sécurité du composant.

NOTE La disponibilité des preuves peut faire l'objet de restrictions d'ordre commercial ou juridique (par exemple, droit d'auteur ou droits de propriété intellectuelle). Ces restrictions ne relèvent pas du domaine d'application de la présente norme.

D.3.3 Lorsque les preuves ne peuvent être fournies pour faciliter l'évaluation de la sécurité fonctionnelle, alors l'élément ne convient pas à une utilisation avec les systèmes E/E/PE relatifs à la sécurité.

--`,`,,`,,`````,,`,`,,`,`,`,`-`-`,,`,,`,`,,`---
Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Annexe E
(informative)

Relation entre la CEI 61508-2 et la CEI 61508-3

Le tableau suivant permet de trouver les articles de la CEI 61508-2 que les personnes concernées par le logiciel uniquement doivent prendre en compte et les articles qu'ils peuvent écarter. Il est bien établi que la quasi-totalité des articles concerne les problèmes liés au matériel. Ce qui n'est par conséquent pas repris dans le présent document. Les aspects importants du logiciel sont traités dans la CEI 61508-3. Cependant, bon nombre des exigences liées au logiciel figurent également dans la CEI 61508-2, venant la plupart du temps recouper les exigences de la CEI 61508-3. La CEI 61508-2 doit être principalement connue des spécialistes en logiciel qui recherchent la compatibilité entre matériel et logiciel. Les exigences de la CEI 61508-2 sont regroupées dans les catégories suivantes:

Tableau E.1 – Catégories des exigences de la CEI 61508-2

Logiciel	Pour les utilisateurs de la norme traitant du matériel et les utilisateurs traitant le logiciel.
Logiciel d'application	Les utilisateurs traitant le logiciel pour résoudre une fonction relative à la sécurité en tant que telle et non pour exploiter le logiciel système ou les fonctions de la bibliothèque.
Logiciel système	Pour les utilisateurs traitant principalement le logiciel système d'exploitation, les fonctions de la bibliothèque et autres.
Matériel uniquement	Ne s'applique pas aux personnes intéressées par le logiciel uniquement.
Matériel principalement	Ne concerne le logiciel que de manière marginale.

Tableau E.2 – Exigences de la CEI 61508-2 pour le logiciel et leur pertinence typique pour certains types de logiciels

Exigence de la CEI 61508-2	Important pour les utilisateurs traitant des éléments suivants	Remarques
7.2	Logiciel	
7.2.3.1	Logiciel d'application	
7.2.3.2 à 7.2.3.6	Logiciel	
7.2.3.3	Matériel uniquement	
7.3	Logiciel	7.3.2.2 f) Matériel uniquement
7.4	Logiciel	
7.4.2.1 à 7.4.2.12	Logiciel	
7.4.2.13, 7.4.2.14	Matériel uniquement	
7.4.3.1 à 7.4.3.3	Logiciel	
7.4.3.4	Matériel uniquement	
7.4.4	Matériel uniquement	
7.4.5	Matériel uniquement	
7.4.6	Logiciel	7.4.6.7 Matériel uniquement
7.4.7	Logiciel	7.4.7.1 a), b) Matériel uniquement
7.4.8	Matériel uniquement	
7.4.9.1 à 7.4.9.3	Logiciel	
7.4.9.4, 7.4.9.5	Matériel uniquement	
7.4.9.6, 7.4.9.7	Logiciel	

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Exigence de la CEI 61508-2	Important pour les utilisateurs traitant des éléments suivants	Remarques
7.4.10	Logiciel	Logiciel système principalement
7.4.11	Matériel uniquement	
7.5	Logiciel	
7.6	Logiciel	
7.6.2.1 a)	Matériel	
7.6.2.4	Matériel principalement	
7.7	Logiciel	7.7.2.3, 7.7.2.4 Logiciel d'application principalement
7.8	Logiciel	
7.9	Logiciel d'application principalement	
8	Logiciel	
Annexe A.1	Matériel principalement	
Annexe A.2 et Tableaux	Matériel principalement	Tableau A.10 Logiciel
Annexe A.3	Matériel principalement	Tableaux A.16, A.17 et A.18 Contiennent certains aspects du logiciel
Annexe B, tous les tableaux	Logiciel	
Annexe C	Matériel	
Annexe D	Logiciel	D.2.3 Matériel uniquement
Annexe E	Matériel uniquement	
Annexe F	Matériel uniquement	

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Annexe F
(informative)

Techniques de réalisation de non interférence entre les composants logiciels d'un seul ordinateur

F.1 Introduction

L'indépendance d'exécution entre les composants logiciels contenus dans un seul système informatique (comprenant un ou plusieurs processeurs avec mémoire et autres matériels partagés avec les processeurs) peut être obtenue et démontrée par plusieurs méthodes différentes. La présente annexe spécifie certaines techniques qu'il est possible d'utiliser pour obtenir la non interférence (entre composants à différentes capabilités systématiques, entre composants conçus pour réaliser ou contribuer à la même fonction de sécurité, ou entre logiciels contribuant à une fonction de sécurité et logiciel non lié à une fonction de sécurité sur le même ordinateur).

NOTE Le terme "indépendance d'exécution" signifie que les composants n'interfèrent pas de manière préjudiciable avec le comportement d'exécution d'autres composants au risque d'engendrer une défaillance dangereuse. Il est utilisé pour différencier d'autres aspects d'indépendance qui peuvent se révéler nécessaires entre composants, notamment la diversité, pour satisfaire à d'autres exigences de la norme.

F.2 Domaines de comportement

Il convient de réaliser et de démontrer l'indépendance d'exécution tant dans le domaine spatial que temporel.

Domaine spatial: les données utilisées par un composant ne doivent pas être modifiées par un autre composant. Notamment, elles ne doivent pas être modifiées par un composant non lié à la sécurité

Domaine temporel: un composant ne doit pas perturber le bon fonctionnement d'un autre composant en prenant une part trop importante de la durée d'exécution du processeur disponible, ou en bloquant l'exécution de l'autre composant en verrouillant une ressource partagée de tout type.

F.3 Analyse des facteurs de causalité

Pour démontrer l'indépendance d'exécution, il convient de réaliser une analyse de la conception proposée pour identifier toutes les causes possibles d'interférence d'exécution entre les composants théoriquement indépendants (non interférents) dans les domaines spatial et temporel. Il convient que l'analyse tienne compte des conditions en fonctionnement normal et en défaillance, et comprennent (sans toutefois s'y limiter) les éléments suivants:

a) utilisation partagée de la mémoire vive;
b) utilisation partagée des périphériques;
c) utilisation partagée du temps de processeur (lorsque deux composants ou plus sont exécutés par un seul processeur);
d) communications entre les composants nécessaires pour réaliser la conception globale;
e) la possibilité qu'une défaillance d'un composant (telle que dépassement, ou exception nulle, ou calcul de pointeur incorrect) puisse engendrer une défaillance dans d'autres composants.

La réalisation et la justification de l'indépendance d'exécution doivent ensuite traiter toutes ces sources d'interférence identifiées.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

F.4 Réalisation de l'indépendance spatiale

Les techniques permettant de réaliser et de démontrer l'indépendance spatiale comprennent les éléments suivants:

a) Utilisation de la protection de mémoire du matériel entre différents composants, y compris les composants à différentes capabilités systématiques.

b) Utilisation d'un système d'exploitation permettant à chaque composant de s'exécuter dans son propre processus avec son propre espace mémoire virtuelle et soutenu par la protection de mémoire du matériel.

c) Utilisation d'une conception rigoureuse, d'un code source et d'une éventuelle analyse de code objet pour démontrer l'absence de toute référence mémoire explicite ou implicite entre les composants logiciels susceptibles d'écraser les données appartenant à un autre composant (en cas d'indisponibilité de la protection de mémoire du matériel).

d) Protection par logiciel des données d'un composant d'intégrité supérieure contre toute modification interdite par un composant d'intégrité inférieure.

Il convient que les données ne soient pas transmises d'un composant d'intégrité inférieure à un composant d'intégrité supérieure sauf si ce dernier peut vérifier que les données présentent un niveau d'intégrité suffisant.

Lorsque les données doivent être transmises entre des composants qui doivent être indépendants, il convient d'utiliser des interfaces unidirectionnelles, telles que des messages ou des canaux de communication, plutôt que la mémoire partagée.

NOTE Théoriquement, les composants indépendants ne devraient pas communiquer entre eux. Cependant, lorsque la conception du système nécessite que les données soient transmises d'un composant à un autre, il convient que le mécanisme de communication interdise toute défaillance ou blocage d'exécution du composant émetteur comme du composant récepteur si la transmission des données est arrêtée ou différée.

Toutes données résidant dans des mémoires permanentes telles que des disques magnétiques, doivent être prises en compte pour le partitionnement spatial, en complément aux données transitoires dans la mémoire vive. Par exemple, la protection d'accès au fichier mise en œuvre par un système d'exploitation peut être utilisée pour empêcher un composant d'écrire dans des zones de données appartenant à un autre composant.

F.5 Réalisation de l'indépendance temporelle

Les techniques permettant de réaliser l'indépendance temporelle comprennent les éléments suivants:

a) Méthodes d'ordonnancement déterministe. Par exemple,

- algorithme d'ordonnancement cyclique qui alloue un créneau temporel défini à chaque composant, soutenu par une analyse de la durée d'exécution la plus défavorable de chaque composant pour démontrer de manière statique la conformité de chaque composant aux exigences de synchronisation;
- architectures à déclenchement temporel.

b) Ordonnancement basé sur une priorité stricte mis en œuvre par un superviseur temps réel avec un moyen d'éviter toute inversion de priorité.

c) Limites de délai qui terminent l'exécution d'un composant s'il dépasse sa durée d'exécution ou délai alloué (dans ce cas, une analyse de danger doit être réalisée pour démontrer que l'achèvement d'un composant n'engendre pas une défaillance dangereuse, cette technique peut de ce fait être mieux appropriée pour un composant non lié à la sécurité).

d) Un système d'exploitation qui garantit qu'aucun processus ne peut être privé de temps de processeur, par exemple, par partage de temps. Une approche de ce type ne peut s'appliquer qu'en l'absence d'exigences de temps réel du matériel auxquelles doivent

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

satisfaire les composants relatifs à la sécurité. Elle indique également que l'algorithme d'ordonnancement ne provoque aucun délai indu pour tout composant.

Lorsqu'une ressource (telle qu'un périphérique) est partagée entre les composants, la conception doit garantir que les composants ne peuvent pas fonctionner de manière incorrecte car la ressource partagée est verrouillée par un autre composant. Le temps d'accès nécessaire à une ressource partagée doit être pris en compte pour déterminer la non interférence temporelle.

F.6 Exigences relatives au logiciel d'aide

Si un système d'exploitation, un superviseur temps réel, un logiciel de gestion de mémoire, un logiciel de gestion de temporisation ou tout autre logiciel de ce type doit être utilisé pour assurer une indépendance spatiale ou temporelle, ou les deux, le logiciel considéré doit alors présenter le niveau de capabilité systématique le plus élevé pour tous les composants devant être indépendants.

NOTE Il est évident que tout logiciel de ce type représente une cause commune potentielle de défaillance des composants indépendants.

F.7 Indépendance des modules logiciels – aspects du langage de programmation

Le Tableau F.1 suivant donne une définition informelle des termes pertinents.

Tableau F.1 – Couplage de modules – définition des termes

Terme	Définition informelle
Cohésion	mesure de l'interdépendance des connexions entre les données et les sous-programmes au sein d'un module
Couplage	mesure de l'interdépendance des connexions entre modules
Encapsulation	masquage des données (privées) internes et des sous-programmes depuis l'accès externe; terme principalement utilisé avec les programmes orientés objet
Indépendance	mesure du découplage des parties du logiciel; complément du couplage
Module	partie de logiciel confinée qui exécute une tâche et qui peut disposer de ses propres données; `Classe`, hiérarchie de classes, sous-programme, `unité, module, paquetage`, … conformément au langage de programmation
Interface	ensemble bien défini de têtes de sous-programmes qui assure l'accès à un module
Donnée de référence irrégulière	donnée qui n'est pas utilisée dans le module récepteur mais qui est uniquement transférée à un autre module

En règle générale, l'indépendance des modules est améliorée s'il existe un couplage lâche entre les modules et une forte cohésion au sein des modules. La forte cohésion favorise les cas où des unités de fonctionnalité identifiables correspondent nettement aux unités identifiables du code d'exécution, tandis que le couplage lâche de module favorise une faible interaction et de ce fait une forte indépendance entre les modules non liés de manière fonctionnelle.

Le couplage lâche de modules permet généralement de réaliser une forte cohésion au sein des modules en regroupant le code et les données utilisés pour réaliser une fonction particulière. Une faible cohésion se produit si le code et les données sont regroupés dans des modules uniquement de manière arbitraire, ou du fait d'une séquence de synchronisation ou encore d'une séquence dans le flux de commande.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Il est possible de différencier plusieurs aspects du couplage de modules, voir le Tableau F.2 ci-dessous.

Tableau F.2 – Types de couplage de modules

Couplage	Définition	Explication	Justification	Remarque
Couplage d'interface, encapsulation	Couplage uniquement par un ensemble bien défini de sous-programmes.	Accès au module ou à ses données uniquement par des sous-programmes; toute modification d'une valeur d'une variable, toute question relative à la valeur de cette variable ou tout autre service requis par le module est acheminé par un appel de sous-programme.	Les têtes des sous-programmes (signatures) d'un module expliquent les services disponibles. Si des modifications d'un module sont nécessaires, la plus grande part de ces modifications peut être réalisée au sein du module considéré sans affecter les autres modules. Favorise le couplage lâche, recommandé de manière générale.	S'applique principalement aux programmes orientés objets, classes, hiérarchies de classes, paquetages de bibliothèques; et non aux sous-programmes.
Couplage de données par liste de paramètres	Transfert de données uniquement par la liste de paramètres ou l'identifiant des sous-programmes.	Accès au module ou à ses données uniquement par des variables ou objets indiqués dans la tête du sous-programme; toute modification d'une valeur d'une variable, toute question relative à la valeur de cette variable est visible.	La tête des sous-programmes présente les données ou objets impliqués par un appel du sous-programme considéré. Favorise le couplage lâche, recommandé de manière générale.	Au sein des classes des programmes orientés objet, ce principe n'est généralement pas observé. L'accès aux variables locales peut être réalisé directement. Une stricte observation de ce principe peut également donner lieu à des données irrégulières. Il convient d'enfreindre ce principe pour éviter ce type de données.
Couplage de structure	Le transfert de données contient plus de données que nécessaire.	Plus de données que nécessaire sont transférées au sous-programme récepteur pour réaliser la fonction requise.	Les données superflues fournissent à un autre module des informations dont il n'a pas besoin pour réaliser sa tâche. Ces données peuvent donner lieu à une mauvaise compréhension de la coopération entre les modules. Ceci n'est cependant pas déconseillé.	Le défaut peut généralement être facilement corrigé.
Couplage de commande	Couplage qui exerce une commande immédiate sur le module récepteur.	Transfert de données qui ne peut provoquer qu'une réaction de branchement dans l'autre module; caractérisé dans la plupart des cas par le transfert d'un seul bit.	Plus strict que les couplages ci-dessus car il nécessite une action immédiate spécifiant au sous-programme récepteur de réagir. A traiter avec précaution; à éviter si possible. Non recommandé de manière générale.	Ne peut pas toujours être évité. Peut se révéler nécessaire, par exemple en cas d'achèvement d'une action ou de validité d'une valeur.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Couplage	Définition	Explication	Justification	Remarque
Couplage global	Couplage par données globales.	Les modules peuvent accéder aux données qui sont directement accessibles par d'autres modules, ou un module peut accéder directement aux données appartenant à un autre module.	Les têtes des sous-programmes n'indiquent pas les données utilisées ni leur source. Il est difficile de comprendre les fonctions des sous-programmes et de prévoir les effets de toute modification apportée au code.	Déconseillé de manière générale. Peut se révéler nécessaire de manière exceptionnelle, par exemple pour éviter des données irrégulières. A utiliser uniquement de manière limitée et ponctuelle pour des normes de codage clairement définies et documentées.
Couplage de contenu	Saut direct dans d'autres modules, influençant les buts du branchement dans d'autres modules, ou accès direct aux données dans d'autres modules.	Réalisable dans les programmes de langage d'assemblage; non réalisable dans tous les langages de niveau supérieur. Peut accélérer l'exécution du programme et réduire l'effort de codage.	Déconseillé. Un seul module peut être compris par la compréhension de ses modules connectés également. Rend un programme extrêmement difficile à comprendre et extrêmement difficile à modifier.	Pas toujours possible dans certains langages de programmation. Peut toujours être évité.

Il convient que la lecture ou la revue de code (voir 7.9.2.12) permette de vérifier que les modules du programme sont ou non à couplage lâche. Cette analyse nécessite généralement d'avoir une certaine compréhension de l'objet des modules et de leur mode de fonctionnement. Un couplage correct ne peut par conséquent être évalué que par la lecture du code et de sa documentation.

Il convient d'éviter le couplage de contenu. Le couplage global peut être utilisé uniquement dans des cas exceptionnels. Il convient d'éviter le couplage de commande et le couplage procédural. Il convient, dans toute la mesure du possible, de connecter les modules par couplage d'interface (encapsulation) et/ou par couplage de données.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Annexe G
(informative)

Indications relatives à la personnalisation des cycles de vie associés aux systèmes dirigés par les données

G.1 Dirigé par les données – partie de système et d'application

De nombreux systèmes sont écrits en deux parties. Une partie fournit la capacité du système de base. L'autre partie adapte le système aux exigences spécifiques de l'application prévue. La partie de l'application peut être écrite sous forme de données qui configurent la partie du système. Dans la présente annexe, ceci est désigné par le terme “dirigé par les données”.

La partie spécifique à l'application du logiciel peut être développée au moyen de divers outils et langages de programmation. Ces langages et outils peuvent imposer des contraintes concernant le mode d'écriture du programme d'application.

Par exemple, lorsqu'un langage de programmation assiste le développeur/configurateur pour décrire la fonctionnalité (par exemple, utilisation d'une logique scalaire pour des systèmes de verrouillage simples), la tâche de programmation du logiciel d’application est alors censée être relativement simple. Cependant, lorsque le langage de programmation permet au développeur/configurateur de décrire un comportement d'application complexe, la tâche de programmation du logiciel d’application peut alors se révéler plus complexe. Dans le cas de développement d'un logiciel d’application très simple, il est possible d'utiliser une conception détaillée pour la configuration plutôt que pour la programmation.

Le niveau de rigueur nécessaire pour réaliser l'intégrité de sécurité requise dépend du niveau de complexité de la configuration disponible pour le développeur/configurateur et de la complexité du comportement à représenter dans l'application. Ceci est illustré de manière schématique sur les axes de la Figure G.1.

Pour des raisons de simplicité, les axes ont été divisés en classes de complexité comme suit:

a) Variabilité autorisée par le langage:
   - programme fixe;
   - variabilité limitée (certaines industries considèrent le programme d'application comme des “données” qui sont interprétées par la partie du système);
   - variabilité totale (bien que généralement non considéré comme dirigé par les données, ce type de système peut également être utilisé pour le développement d'application et figure dans la présente annexe pour ce qui concerne la complétude).

b) Aptitude à configurer l'application:
   - limitée;
   - totale.

Dans la réalité, un système particulier peut présenter différents niveaux de complexité et de configurabilité. Par ailleurs, la complexité peut présenter une échelle variable le long du continuum des deux axes. S'agissant de la personnalisation du cycle de vie du logiciel, il convient d'identifier le niveau de complexité applicable et de justifier le niveau de personnalisation.

Une description des types généraux de système pour chaque niveau de complexité est donnée ci-dessous. La CEI 61508-7 donne des indications sur les techniques proposées pour mettre en œuvre chaque type de système.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

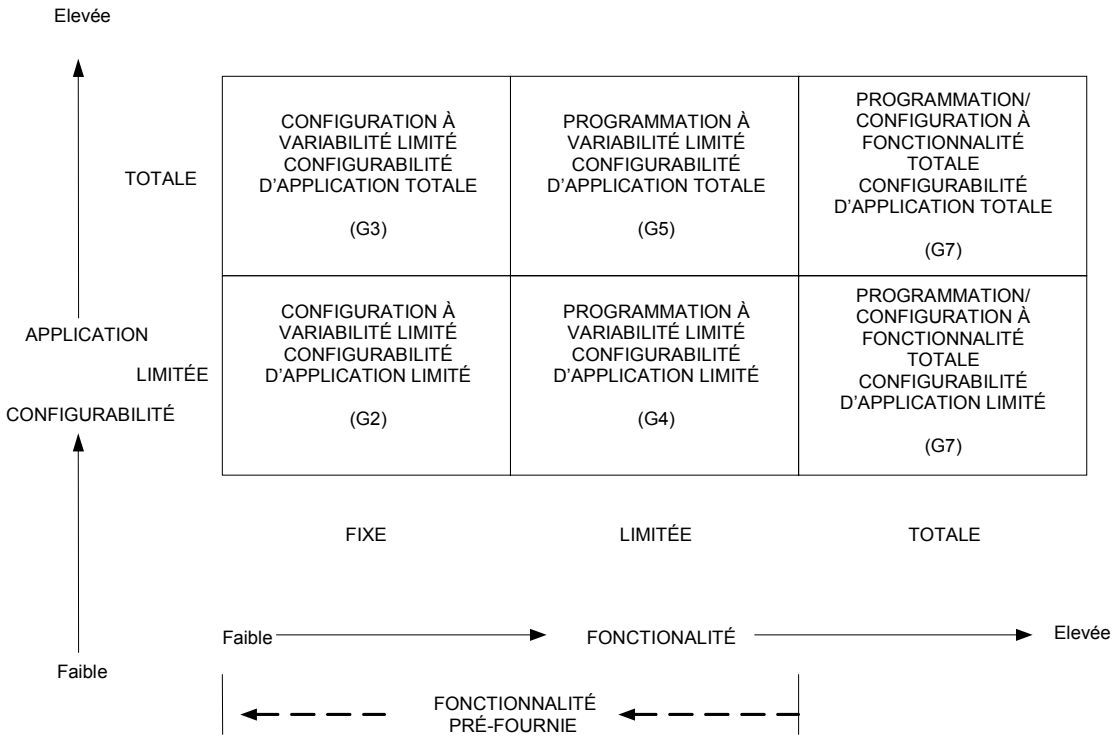


Figure G.1 – Variabilité de complexité des systèmes dirigés par les données

Des systèmes types de chaque classe de complexité sont décrits en G.2.

G.2 Configuration à variabilité limitée, configurabilité d'application limitée

Langage de configuration propriétaire utilisé avec un système conforme à la CEI 61508 avec fonctionnalité pré-fournie fixe.

Le langage de configuration ne permet pas au programmeur de modifier la fonction du système. En revanche, la configuration est limitée au réglage de quelques paramètres (des données) pour pouvoir adapter le système à son application. Des exemples peuvent comprendre les capteurs et actionneurs intelligents après saisie de paramètres spécifiques, contrôleurs de réseau, séquenceurs, petits systèmes d'enregistrement des données et instruments intelligents.

Il convient que la justification de la personnalisation du cycle de vie de sécurité comprenne, sans toutefois s'y limiter, les éléments suivants:

a) spécification des paramètres d'entrée pour cette application;

b) vérification de la bonne mise en œuvre des paramètres dans le système d'exploitation;

c) validation de toutes les combinaisons des paramètres d'entrée;

d) prise en compte des modes de fonctionnement spéciaux et spécifiques pendant la configuration;

e) facteurs humains / ergonomie;

f) dispositifs de verrouillage, par exemple, assurer que les dispositifs de verrouillage d'exploitation ne sont pas invalidés pendant le processus de configuration;

--`,``,,`````````,,``,`,,``,`,`-`-`,,`,,`,`,,`---

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

g) reconfiguration accidentelle, par exemple, accès par interrupteur à clé, dispositifs de protection.

G.3 Configuration à variabilité limitée, configurabilité d'application totale

Langage de configuration propriétaire utilisé avec un système conforme à la CEI 61508 avec fonctionnalité pré-fournie fixe.

Le langage de configuration ne permet pas au programmeur de modifier la fonction du système. En revanche, la configuration est contrainte de créer des paramètres de données statiques étendus pour pouvoir adapter le système à son application. Un exemple peut être un système de contrôle de la circulation aérienne comprenant des données avec un grand nombre d'entités de données, chacune d'elles comportant un ou plusieurs attributs. Une caractéristique essentielle des données réside dans le fait qu'elles ne comportent aucun séquencement explicite ni éléments d'ordonnancement ou de branchement et ne contiennent aucune représentation des états combinatoires de l'application.

En complément aux considérations données en G.2, il convient que la justification de la personnalisation du cycle de vie de sécurité comprenne, sans toutefois s'y limiter, les éléments suivants:

a) outils d'automatisation pour la création des données;

b) contrôle de cohérence, par exemple, les données sont auto-compatibles;

c) contrôle des règles, par exemple, pour garantir la conformité de la génération des données aux contraintes définies;

d) validité des interfaces avec les systèmes de préparation des données.

G.4 Programmation à variabilité limitée, configurabilité d'application limitée

Langage orienté problème, utilisé avec un système conforme à la CEI 61508, où les instructions de langage comportent ou s'apparentent à la terminologie de l'application de l'utilisateur pour les systèmes à fonctionnalité pré-fournie limitée.

Ces langages permettent à l'utilisateur d'appliquer une flexibilité limitée à la personnalisation des fonctions du système à ses propres exigences spécifiques, sur la base d'une gamme de matériels et de composants logiciels.

Une caractéristique essentielle de la programmation à variabilité limitée réside dans le fait que les données peuvent comporter des éléments de séquencement, d'ordonnancement ou de branchement explicites et peuvent invoquer des états combinatoires de l'application. Des exemples peuvent comprendre la programmation de blocs fonctionnels, la logique scalaire, les tableurs et les systèmes graphiques.

En complément aux considérations données en G.3, il convient d'inclure, sans toutefois s'y limiter, les éléments suivants:

a) la spécification des exigences d'application;

b) les sous-ensembles de langage autorisés pour cette application;

c) les méthodes de conception permettant de combiner les sous-ensembles de langage;

d) les critères de couverture pour la vérification concernant les combinaisons des états du système potentiels.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

G.5 Programmation à variabilité limitée, configurabilité d'application totale

Langage orienté problème, utilisé avec un système conforme à la CEI 61508, où les instructions de langage comportent ou s'apparentent à la terminologie de l'application de l'utilisateur pour les systèmes à fonctionnalité pré-fournie limitée.

La différence essentielle avec la programmation à variabilité limitée et la configurabilité d'application limitée est la complexité de la configuration de l'application. Des exemples peuvent comprendre les systèmes graphiques et les systèmes de contrôle par lot SCADA.

En complément aux considérations données en G.4, il convient d'inclure, sans toutefois s'y limiter, les éléments suivants:

a) la conception architecturale de l'application;

b) la disposition relative aux modèles;

c) la vérification des modèles individuels;

d) la vérification et la validation de l'application.

L'aspect du cycle de vie donné dans la présente norme le plus susceptible de se révéler non nécessaire (en fonction du langage utilisé) est la mise en œuvre et les essais de modules de niveau inférieur.

G.6 Programmation/configuration à fonctionnalité totale, configurabilité d'application limitée

Voir G.7 ci-dessous.

G.7 Programmation/configuration à fonctionnalité totale, configurabilité d'application totale

Pour ces systèmes, toutes les exigences relatives au cycle de vie spécifiées dans la présente norme s'appliquent.

Les parties des systèmes à variabilité totale sont basées sur des langages de programmation d'usage général ou des langages de base de données d'usage général ou encore sur des suites de logiciels scientifiques et de simulation d'usage général. Ces parties sont généralement utilisées conjointement à un système informatique avec un système d'exploitation qui assure l'allocation des ressources système et un environnement multiprogrammation en temps réel. Des exemples de systèmes susceptibles d'être écrits dans des langages à variabilité totale peuvent comprendre par exemple: un système de commande d'automate dédié, les systèmes de commande de vol spécialisés ou les services web pour la gestion des services relatifs à la sécurité.

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Bibliographie

[1] CEI 61511 (toutes les parties), *Sécurité fonctionnelle – Systèmes instrumentés de sécurité pour le secteur des industries de transformation*

[2] CEI 62061, *Sécurité des machines – Sécurité fonctionnelle des systèmes de commande électriques, électroniques et électroniques programmables relatifs à la sécurité*

[3] CEI 61800-5-2, *Adjustable speed electrical power drive systems – Part 5-2: Safety requirements – Functional* (disponible en anglais seulement)

[4] CEI 61508-5: 2010, *Sécurité fonctionnelle des systèmes électriques/électroniques/ électroniques programmables relatifs à la sécurité – Partie 5: Exemples de méthodes pour la détermination des niveaux d'intégrité de sécurité*

[5] CEI 61508-6: 2010, *Sécurité fonctionnelle des systèmes électriques / électroniques / électroniques programmables relatifs à la sécurité – Partie 6: Lignes directrices pour l'application de la CEI 61508-2 et de la CEI 61508-3*

[6] CEI 61508-7 : 2010, *Sécurité fonctionnelle des systèmes électriques / électroniques / électroniques programmables relatifs à la sécurité – Partie 7: Présentation de techniques et mesures*

[7] CEI 60601 (toutes les parties), *Appareils électromédicaux*

[8] CEI 61131-3, *Programmable controllers – Part 3: Programming languages* (disponible en anglais seulement)

_____

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS
Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch

Copyright International Electrotechnical Commission
Provided by IHS under license with IEC
No reproduction or networking permitted without license from IHS

Licensee=Rockwell Automation/9964823100, User=Hanneman, Jeff
Not for Resale, 09/22/2010 05:37:45 MDT