# IEC 62304

Edition 1.0 2015-06

# INTERNATIONAL STANDARD

# NORME INTERNATIONALE

AMENDMENT 1
AMENDEMENT 1

## Medical device software – Software life cycle processes

## Logiciels de dispositifs médicaux – Processus du cycle de vie du logiciel



IEC 62304:2006-05/AMD1:2015-06(en-fr)

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

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

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

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

**IEC Catalogue - webstore.iec.ch/catalogue**
The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

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 It also gives information on projects, replaced and withdrawn publications.

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

**Electropedia - www.electropedia.org**
The world's leading online dictionary of electronic and electrical terms containing 20 000 terms and definitions in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

**IEC Glossary - std.iec.ch/glossary**
65 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

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

---

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

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

**Catalogue IEC - webstore.iec.ch/catalogue**
Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

**Recherche de publications IEC - www.iec.ch/searchpub**
La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,…). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

**IEC Just Published - webstore.iec.ch/justpublished**
Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

**Electropedia - www.electropedia.org**
Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient 20 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

**Glossaire IEC - std.iec.ch/glossary**
65 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

**Service Clients - webstore.iec.ch/csc**
Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

# IEC 62304

Edition 1.0 2015-06

# INTERNATIONAL STANDARD

# NORME INTERNATIONALE

AMENDMENT 1
AMENDEMENT 1

## Medical device software – Software life cycle processes

## Logiciels de dispositifs médicaux – Processus du cycle de vie du logiciel



INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 11.040

ISBN 978-2-8322-4638-2

# FOREWORD

This amendment has been prepared by a joint working group of subcommittee 62A: Common aspects of electrical equipment used in medical practice, of IEC technical committee 62: Electrical equipment in medical practice and ISO Technical Committee 210, Quality management and corresponding general aspects for MEDICAL DEVICES.

This publication is published as a double logo standard.

This bilingual version (2017-09) corresponds to the monolingual English version, published in 2015-06.

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

| FDIS | Report on voting |
|---|---|
| 62A/1007/FDIS | 62A/1014/RVD |

Full information on the voting for the approval of this amendment can be found in the report on voting indicated in the above table. In ISO, the standard has been approved by 30 P-members out of 30 having cast a vote.

The French version of this amendment has not been voted upon.

The committee has decided that the contents of this amendment and the base publication will remain unchanged until the stability date indicated on the IEC website 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.



---

# INTRODUCTION TO THE AMENDMENT 1

The first edition of IEC 62304 was published in 2006. This amendment is intended to add requirements to deal with LEGACY SOFTWARE, where the software design is prior to the existence of the current version, to assist manufacturers who must show compliance to the standard to meet European Directives. Software safety classification changes needed for this amendment include clarification of requirements and updating of the software safety classification to include a risk-based approach. Work is continuing in parallel to develop the second edition of IEC 62304.

## FOREWORD

*Add the following note at the end of the Foreword:*

NOTE The attention of National Committees is drawn to the fact that equipment MANUFACTURERS and testing organizations may need a transitional period following publication of a new, amended or revised IEC or ISO publication in which to make products in accordance with the new requirements and to equip themselves for conducting new or revised tests. It is the recommendation of the committee that the content of this publication be adopted for mandatory implementation nationally not earlier than 3 years from the date of publication.

## INTRODUCTION

*Replace, in the second paragraph, the existing third sentence with the following:*

Each life cycle PROCESS consists of a set of ACTIVITIES, with most ACTIVITIES consisting of a set of TASKS.

*Replace, in the first sentence of the fourth paragraph, the phrase "*contributing factor to a HAZARD*" with "* contributing factor to a HAZARDOUS SITUATION*".*

*Replace, in the second sentence of the fourth paragraph, the term,* "HAZARDS" *with* "HAZARDOUS SITUATIONS".

*Add, after the existing sixth paragraph, the following new paragraph:*

Amendment 1 updates the standard to add requirements to deal with LEGACY SOFTWARE, where the software design is prior to the existence of the current version, to assist manufacturers who must show compliance to the standard to meet European Directives. Software safety classification changes include clarification of requirements and updating of the software safety classification to include a risk-based approach.

## 1 Scope

### 1.2 * Field of application

*Replace the entire existing text of this subclause with the following:*

This standard applies to the development and maintenance of MEDICAL DEVICE SOFTWARE when software is itself a MEDICAL DEVICE or when software is an embedded or integral part of the final MEDICAL DEVICE.

NOTE 1 This standard can be used in the development and maintenance of software that is itself a medical device. However, additional development activities are needed at the system level before this type of software can be placed into service. These system activities are not covered by this standard, but can be found in IEC 82304-1[1] [22].

This standard describes PROCESSES that are intended to be applied to software which executes on a processor or which is executed by other software (for example an interpreter) which executes on a processor.

This standard applies regardless of the persistent storage device(s) used to store the software (for example: hard disk, optical disk, permanent or flash memory).

This standard applies regardless of the method of delivery of the software (for example: transmission by network or email, optical disk, flash memory or EEPROM). The method of software delivery itself is not considered MEDICAL DEVICE SOFTWARE.

This standard does not cover validation and final release of the MEDICAL DEVICE, even when the MEDICAL DEVICE consists entirely of software.

NOTE 2 If a medical device incorporates embedded software intended to be executed on a processor, the requirements of this standard apply to the software, including the requirements concerning software of unknown provenance (see 8.1.2).

NOTE 3 Validation and other development activities are needed at the system level before the software and medical device can be placed into service. These system activities are not covered by this standard, but can be found in related product standards (e.g., IEC 60601-1, IEC 82304-1, etc.).

### 1.4 Compliance

*Delete, in the second paragraph, the instruction* "See Annex D."

*Add, after existing Note 4, the following new note:*

NOTE 5 For compliance of LEGACY SOFTWARE see 4.4.



[1] In preparation.

## 3 * Terms and definitions

**3.2**
**ANOMALY**

*Replace, in the definition,* "SOFTWARE PRODUCTS" *with* "MEDICAL DEVICE SOFTWARE".

*Replace the existing source reference with the following note:*

NOTE Based on IEEE 1044:1993, definition 3.1.

**3.4**
**CHANGE REQUEST**

*Replace* "SOFTWARE PRODUCT" *with* "MEDICAL DEVICE SOFTWARE".

**3.5**
**CONFIGURATION ITEM**

*Replace, in the note,* "ISO/IEC 12207:1995, definition 3.6" *with* "ISO/IEC 12207:2008, 4.7".

**3.7**
**EVALUATION**

*Replace the existing source reference with* "[ISO/IEC 12207:2008, 4.12]".

**3.8**
**HARM**

*Replace the existing source reference with* "[ISO 14971:2007, 2.2]".

**3.9**
**HAZARD**

*Replace the existing source reference with* "[ISO 14971:2007, 2.3]".

**3.10**
**MANUFACTURER**

*Add the following new notes:*

NOTE 1 Attention is drawn to the fact that the provisions of national or regional regulations can apply to the definition of manufacturer.

NOTE 2 For a definition of labelling, see ISO 13485:2003, definition 3.6.

*Replace the existing source reference with* "[ISO 14971:2007, 2.8]".

**3.11**
**MEDICAL DEVICE**

*Add the following new note:*

NOTE 3 In conjunction with IEC 60601-1:2005 and IEC 60601-1:2005/AMD1:2012 the term "medical device" assumes the same meaning as ME EQUIPMENT or ME SYSTEM (which are defined terms of IEC 60601-1).

**3.12**
**MEDICAL DEVICE SOFTWARE**

*Replace the existing definition with the following:*

SOFTWARE SYSTEM that has been developed for the purpose of being incorporated into the MEDICAL DEVICE being developed or that is intended for use as a MEDICAL DEVICE

NOTE This includes a MEDICAL DEVICE software product, which then is a MEDICAL DEVICE in its own right.

**3.13**
**PROBLEM REPORT**

*Replace, in the definition and in Notes 1 and 2,* "SOFTWARE PRODUCT" *with* "MEDICAL DEVICE SOFTWARE" *(4 times)*.

**3.16**
**RISK**

*Replace the existing source reference with* "[ISO 14971:2007, 2.16]".

**3.17**
**RISK ANALYSIS**

*Replace the existing source reference with* "[ISO 14971:2007, 2.17]".

**3.18**
**RISK CONTROL**

*Replace the existing source reference with* "[ISO 14971:2007, 2.19]".

**3.19**
**RISK MANAGEMENT**

*Replace the existing source reference with* "[ISO 14971:2007, 2.22, modified – The phrase "and monitoring" has been removed]".

**3.20**
**RISK MANAGEMENT FILE**

*Replace the existing source reference with* "[ISO 14971:2007, 2.23]".

**3.21**
**SAFETY**

*Replace the existing source reference with* "[ISO 14971:2007, 2.24]".

**3.22**
**SECURITY**

*Replace the existing definition with the following:*

protection of information and data so that unauthorized persons or systems cannot read or modify them an authorized persons or systems are not denied access to them.

NOTE Based on ISO/IEC 12207:2008, 4.39.

**3.23**
**SERIOUS INJURY**

*Delete, in the first line of the definition, the words* "directly or indirectly".

**3.24**
**SOFTWARE DEVELOPMENT LIFE CYCLE MODEL**

*Delete, in the second line of the definition, the phrase "for* manufacturing*".*

*Replace, in the first dashed item, the words* "a SOFTWARE PRODUCT" *with* "MEDICAL DEVICE SOFTWARE".

**3.25**
**SOFTWARE ITEM**

*Replace the existing definition with the following:*

any identifiable part of a computer program, i.e., source code, object code, control code, control data, or a collection of these items

NOTE 1 Three terms identify the software decomposition. The top level is the SOFTWARE SYSTEM. The lowest level that is not further decomposed is the SOFTWARE UNIT. All levels of composition, including the top and bottom levels, can be called SOFTWARE ITEMS. A SOFTWARE SYSTEM, then, is composed of one or more SOFTWARE ITEMS, and each SOFTWARE ITEM is composed of one or more SOFTWARE UNITS or decomposable SOFTWARE ITEMS. The responsibility is left to the MANUFACTURER to provide the granularity of the SOFTWARE ITEMS and SOFTWARE UNITS.

NOTE 2 Based on ISO/IEC 90003:2004, 3.14 and ISO/IEC 12207:2008, 4.41.

**3.26**
**SOFTWARE PRODUCT**

*Delete the existing term and definition and add "Not used".*

**3.28**
**SOFTWARE UNIT**

*Replace the existing note by the following:*

NOTE The granularity of SOFTWARE UNITS is defined by the MANUFACTURER (see B.3).

**3.29**
**SOUP**
**software of unknown provenance (acronym)**

*Replace, in the third line of the definition,* "software previously developed" *with* "SOFTWARE ITEM previously developed"

*Add the following new note:*

NOTE A MEDICAL DEVICE SOFTWARE SYSTEM in itself cannot be claimed to be SOUP.

**3.30**
**SYSTEM**

*Replace the existing source reference with the following note:*

NOTE Based on ISO/IEC 12207:2008, 4.48.

**3.32**
**TRACEABILITY**

*Add the following new note:*

NOTE Requirements, architecture, risk control measures, etc. are examples of deliverables of the development PROCESS.

**3.34**
**VERSION**

*Replace, in the existing text of Note1, the words* "a SOFTWARE PRODUCT" *with* "MEDICAL DEVICE SOFTWARE".

*Replace the existing text of Note 2 with the following*

NOTE 2 Based on ISO/IEC 12207:2008, 4.56.

*Add the following new definitions:*

**3.35**
**HAZARDOUS SITUATION**
circumstance in which people, property or the environment are exposed to one or more HAZARD(S)

[SOURCE: ISO 14971:2007, 2.4]

**3.36**
**LEGACY SOFTWARE**
MEDICAL DEVICE SOFTWARE which was legally placed on the market and is still marketed today but for which there is insufficient objective evidence that it was developed in compliance with the current version of this standard

**3.37**
**RELEASE**
particular VERSION of a CONFIGURATION ITEM that is made available for a specific purpose

NOTE Based on ISO/IEC 12207:2008, definition 4.35.

**3.38**
**RESIDUAL RISK**
RISK remaining after RISK CONTROL measures have been taken

NOTE 1 Adapted from ISO/IEC Guide 51:1999, definition 3.9.

NOTE 2 ISO/IEC Guide 51:1999, definition 3.9 uses the term "protective measures" rather than "RISK CONTROL measures." However, in the context of this International Standard, "protective measures" are only one option for controlling RISK as described in 6.2 [of ISO 14971:2007].

[SOURCE: ISO 14971:2007, 2.15].

**3.39**
**RISK ESTIMATION**
PROCESS used to assign values to the probability of occurrence of HARM and the severity of that HARM

[SOURCE: ISO 14971:2007 2.20]

**3.40**
**RISK EVALUATION**
PROCESS of comparing the estimated RISK against given RISK criteria to determine the acceptability of the RISK

[SOURCE: ISO 14971:2007 2.21]

### 4.3 * Software safety classification

*Replace existing items a) and b), and insert a new Figure 3, as follows:*

a) The MANUFACTURER shall assign to each SOFTWARE SYSTEM a software safety class (A, B, or C) according to the RISK of HARM to the patient, operator, or other people resulting from a HAZARDOUS SITUATION to which the SOFTWARE SYSTEM can contribute in a worst-case-scenario as indicated in Figure 3.

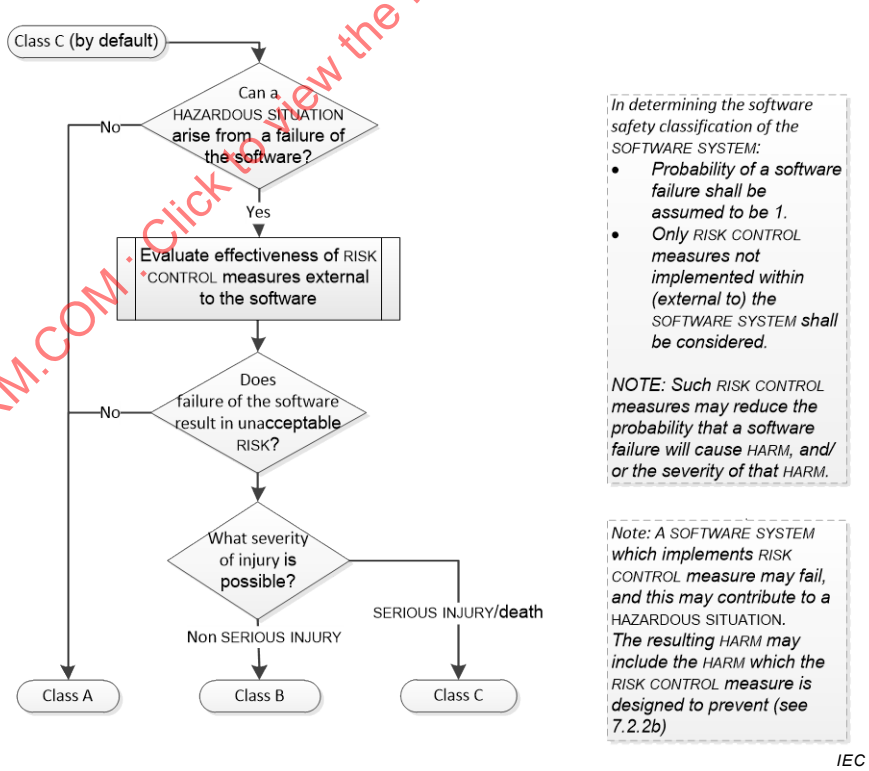



**Figure 3 – Assigning software safety classification**

The SOFTWARE SYSTEM is software safety class A if:

– the SOFTWARE SYSTEM cannot contribute to a HAZARDOUS SITUATION; or

– the SOFTWARE SYSTEM can contribute to a HAZARDOUS SITUATION which does not result in unacceptable RISK after consideration of RISK CONTROL measures external to the SOFTWARE SYSTEM.

The SOFTWARE SYSTEM is software safety class B if:

– the SOFTWARE SYSTEM can contribute to a HAZARDOUS SITUATION which results in unacceptable RISK after consideration of RISK CONTROL measures external to the SOFTWARE SYSTEM and the resulting possible HARM is non-SERIOUS INJURY.

The SOFTWARE SYSTEM is software safety class C if:

– the SOFTWARE SYSTEM can contribute to a HAZARDOUS SITUATION which results in unacceptable RISK after consideration of RISK CONTROL measures external to the SOFTWARE SYSTEM and the resulting possible HARM is death or SERIOUS INJURY.

For a SOFTWARE SYSTEM initially classified as software safety class B or C, the MANUFACTURER may implement additional RISK CONTROL measures external to the SOFTWARE SYSTEM (including revising the system architecture containing the SOFTWARE SYSTEM) and subsequently assign a new software safety classification to the SOFTWARE SYSTEM.

NOTE 1 External RISK CONTROL measures can be hardware, an independent SOFTWARE SYSTEM, health care procedures, or other means to minimize that software can contribute to a HAZARDOUS SITUATION.

NOTE 2 See ISO 14971:2007 subclause 3.2, *Management Responsibilities*, for the definition of risk acceptability.

b) Not used.

*Add, at the end of the first sentence in item d), the following parenthetical phrase: "*(software safety classes assigned according to 4.3 a) replacing "SOFTWARE SYSTEM" with "SOFTWARE ITEM")".

*Replace the existing text of item f) with the following:*

f) For compliance with this standard, when applying this standard to a group of SOFTWARE ITEMS, the MANUFACTURER shall use the PROCESSES and TASKS which are required by the classification of the highest-classified SOFTWARE ITEM in the group unless the MANUFACTURER documents in the RISK MANAGEMENT FILE a rationale for using a lower classification.

*Replace the existing text of the note with the following:*

NOTE In the clauses and subclauses that follow, the software safety classes for which a specific requirement applies are identified following the requirement in the form of [Class . . .].

*Add the following new subclause:*

### 4.4 * LEGACY SOFTWARE

#### 4.4.1 General

As an alternative to applying Clauses 5 through 9 of this standard, compliance of LEGACY SOFTWARE may be demonstrated as indicated in 4.4.2 to 4.4.5.

#### 4.4.2 RISK MANAGEMENT ACTIVITIES

In accordance with 4.2 of this standard, the MANUFACTURER shall:

a) assess any feedback, including post-production information, on LEGACY SOFTWARE regarding incidents and / or near incidents, both from inside its own organization and / or from users;

b) perform RISK MANAGEMENT ACTIVITIES associated with continued use of the LEGACY SOFTWARE, considering the following aspects:

– integration of the LEGACY SOFTWARE in the overall MEDICAL DEVICE architecture;

– continuing validity of RISK CONTROL measures, implemented as part of the LEGACY SOFTWARE;

- identification of HAZARDOUS SITUATIONS associated with the continued use of the LEGACY SOFTWARE;
- identification of potential causes of the LEGACY SOFTWARE contributing to a HAZARDOUS SITUATION;
- definition of RISK CONTROL measures for each potential cause of the LEGACY SOFTWARE contributing to a HAZARDOUS SITUATION.

#### 4.4.3 Gap analysis

Based on the software safety class of the LEGACY SOFTWARE (see 4.3), the MANUFACTURER shall perform a gap analysis of available DELIVERABLES against those required according to 5.2, 5.3, 5.7, and Clause 7.

a) The MANUFACTURER shall assess the continuing validity of available DELIVERABLES.

b) Where gaps are identified, the MANUFACTURER shall EVALUATE the potential reduction in RISK resulting from the generation of the missing DELIVERABLES and associated ACTIVITIES.

c) Based on this evaluation, the MANUFACTURER shall determine the DELIVERABLES to be created and associated ACTIVITIES to be performed. The minimum DELIVERABLE shall be SOFTWARE SYSTEM test records (see 5.7.5).

NOTE Such gap analysis should assure that RISK CONTROL measures, implemented in LEGACY SOFTWARE, are included in the software requirements.

#### 4.4.4 Gap closure activities

a) The MANUFACTURER shall establish and execute a plan to generate the identified DELIVERABLES. Where available, objective evidence may be used to generate required DELIVERABLES without performing ACTIVITIES required by 5.2, 5.3, 5.7 and Clause 7.

NOTE A plan on how to address the identified gaps can be included in a software maintenance plan (see 6.1).

b) The plan shall address the use of the problem resolution PROCESS for handling problems detected in the LEGACY SOFTWARE and DELIVERABLES in accordance with Clause 9.

c) Changes to the LEGACY SOFTWARE shall be performed in accordance with Clause 6.

#### 4.4.5 Rationale for use of LEGACY SOFTWARE

The MANUFACTURER shall document the VERSION of the LEGACY SOFTWARE together with a rationale for the continued use of the LEGACY SOFTWARE based on the outputs of 4.4.

NOTE Fulfilling 4.4 enables further use of LEGACY SOFTWARE in accordance with IEC 62304.

## 5 Software development PROCESS

### 5.1 * Software development planning

#### 5.1.1 Software development plan

*Replace, in list item e),* "SOFTWARE PRODUCTS" *with* "MEDICAL DEVICE SOFTWARE".

#### 5.1.3 Software development plan reference to SYSTEM design and development

*Replace the existing text of list item b) with the following:*

b) In the software development plan, the MANUFACTURER shall include or reference procedures for coordinating the software development with the system development necessary to satisfy 4.1 (such as system integration, verification, and validation).

#### 5.1.5 Software integration and integration testing planning

*Add the following new note and renumber the first note as Note 1:*

NOTE 2 See 5.6.

#### 5.1.8 Documentation planning

*Delete list item c).*

*Add the following new note:*

NOTE See Clause 8 for consideration of configuration management of documentation.

### 5.1.9 Software configuration management planning

*Replace, in list item c),* "software configuration management and ACTIVITIES" *with* "software configuration management ACTIVITIES".

*Add the following new note:*

NOTE See Clause 8.

### 5.1.10 Supporting items to be controlled

*Add the following new note and renumber the first note as NOTE 1:*

NOTE 2 See Clause 8.

### 5.1.11 Software CONFIGURATION ITEM control before VERIFICATION

*Replace* "under documented configuration management" *with* "under configuration management".

*Add the following new subclause:*

### 5.1.12 Identification and avoidance of common software defects

The MANUFACTURER shall include or reference in the software development plan a procedure for:

a) identifying categories of defects that may be introduced based on the selected programming technology that are relevant to their SOFTWARE SYSTEM; and

b) documenting evidence that demonstrates that these defects do not contribute to unacceptable RISK.

NOTE See Annex B of IEC TR 80002-1:2009 for examples of categories of defects or causes contributing to HAZARDOUS SITUATIONS.

[Class B, C]

## 5.2 * Software requirements analysis

### 5.2.2 Software requirements content

*Add to the bulleted list of examples in Note 3 the following additional item;*

– system security/malware protection.

*Replace the existing text of list item f) with the following:*

f) user interface requirements implemented by software;

*Replace, in Note 5* "IEC 60601-1-6." *with* "IEC 62366-1 [21] among others (e.g., IEC 60601-1-6 [3])."

*Replace the existing text of list item j) with the following new text and note:*

j) requirements related to IT-network aspects;

NOTE 9 Examples include those related to:

– networked alarms, warnings, and operator messages;

– network protocols;

– handling of unavailability of network services.

*Add the following new note after list item l):*

NOTE 10 The requirements in a) through l) can overlap.

*Replace, in existing Note 8,* "ISO/IEC 9126-1 [8]" with "Among others, ISO/IEC 25010 [12]"

### 5.2.3 Include RISK CONTROL measures in software requirements

*Delete the phrase* "for hardware failures and potential software defects".

### 5.2.5 Update system requirements

*Replace the existing title of the subclause with the following:*

### 5.2.5 Update requirements

### 5.2.6 Verify software requirements

*Delete, in the existing text of list item d) the phrase* "to determine whether the test criteria have been met".

### 5.3 Software ARCHITECTURAL design

### 5.3.5 Identify segregation necessary for RISK CONTROL

*Replace the existing text of the subclause with the following:*

The MANUFACTURER shall identify any segregation between SOFTWARE ITEMS that is necessary for RISK CONTROL, and state how to ensure that such segregation is effective. [Class C]

NOTE An example of segregation is to have SOFTWARE ITEMS execute on different processors. The effectiveness of the segregation can be ensured by having no shared resources between the processors. Other means of segregation can be applied when effectiveness can be ensured by the software ARCHITECTURE design (see B.4.3).

### 5.3.6 Verify software ARCHITECTURE

*Add the following new note:*

NOTE A TRACEABILITY analysis of ARCHITECTURE to software requirements can be used to satisfy requirement a).

### 5.4 * Software detailed design

### 5.4.1 Refine SOFTWARE ARCHITECTURE into SOFTWARE UNITS

*Replace the existing title and text of this subclause with the following:*

### 5.4.1 Subdivide software into SOFTWARE UNITS

The MANUFACTURER shall subdvide the software until it is represented by SOFTWARE UNITS. [Class B, C]

NOTE Some SOFTWARE SYSTEMS are not divided further.

### 5.4.2 Develop detailed design for each SOFTWARE UNIT

*Replace the existing text with the following:*

The MANUFACTURER shall document a design with enough detail to allow correct implementation of each SOFTWARE UNIT. [Class C]

### 5.4.3 Develop detailed design for interfaces

*Replace the existing text with the following:*

The MANUFACTURER shall document a design for any interfaces between the SOFTWARE UNIT and external components (hardware or software), as well as any interfaces between SOFTWARE UNITS, detailed enough to implement each SOFTWARE UNIT and its interfaces correctly. [Class C]

### 5.4.4 VERIFY detailed design

*Add the following new note:*

NOTE It is acceptable to use a TRACEABILITY analysis of ARCHITECTURE to software detailed design to satisfy requirement a).

### 5.5 SOFTWARE UNIT implementation and verification

*Replace the existing title of this subclause with the following:*

### 5.5 SOFTWARE UNIT implementation

### 5.5.2 Establish SOFTWARE UNIT VERIFICATION PROCESS

*Replace the existing text with the following:*

The MANUFACTURER shall establish strategies, methods and procedures for verifying the SOFTWARE UNITS. Where VERIFICATION is done by testing, the test procedures shall be EVALUATED for adequacy. [Class B, C]

#### 5.5.3 SOFTWARE UNIT acceptance criteria

*Replace the existing text of the second dashed item of the note with the following:*

– is the software code free from contradiction with the interface design of the SOFTWARE UNIT?

### 5.6 Software integration and integration testing

#### 5.6.2 Verify software integration

*Replace the existing text with the following:*

The MANUFACTURER shall verify that the SOFTWARE UNITS have been integrated into SOFTWARE ITEMS and/or the SOFTWARE SYSTEM in accordance with the integration plan (see 5.1.5) and retain records of the evidence of such verification. [Class B, C]

NOTE This VERIFICATION is only that the integration has been done according to the plan. This VERIFICATION is most likely implemented by some form of inspection.

#### 5.6.3 Test integrated software

*Replace the existing title with the following*

#### 5.6.3 Software integration testing

#### 5.6.4 Integration testing content

*Replace the existing title with the following:*

#### 5.6.4 Software integration testing content

#### 5.6.5 Verify integration test procedures

*Replace the existing title and text with*

#### 5.6.5 EVALUATE software integration test procedures

The MANUFACTURER shall EVALUATE the integration test procedures for adequacy. [Class B, C]

### 5.7 SOFTWARE SYSTEM testing

#### 5.7.1 Establish tests for software requirements

*Designate the existing text of the subclause as list item a).*

*Replace "*[Class B, C]*" with "*[Class A, B, C]*".*

*Add the following new list item:*

b) The MANUFACTURER shall EVALUATE the adequacy of VERIFICATION strategies and test procedures.

#### 5.7.2 Use software problem resolution PROCESS

*Replace "*[Class B, C]*" with "*[Class A, B, C]*".*

#### 5.7.3 Retest after changes

*Replace "*[Class B, C]*" with "*[Class A, B, C]*".*

#### 5.7.4 Verify SOFTWARE SYSTEM testing

*Replace the existing title and text of this subclause with:*

#### 5.7.4 EVALUATE SOFTWARE SYSTEM testing

The MANUFACTURER shall EVALUATE the appropriateness of VERIFICATION strategies and test procedures.

The MANUFACTURER shall verify that:

a) all software requirements have been tested or otherwise VERIFIED;

b) the TRACEABILITY between software requirements and tests or other VERIFICATION is recorded; and

c) test results meet the required pass/fail criteria.

[Class A, B, C]

#### 5.7.5 SOFTWARE SYSTEM test record contents

*Replace the existing text with:*

In order to support the repeatability of tests, the MANUFACTURER shall document:

a) a reference to test case procedures showing required actions and expected results;

b) the test result (pass/fail and a list of ANOMALIES);

c) the version of software tested;

d) relevant hardware and software test configurations;

e) relevant test tools;

f) date tested; and

g) the identity of the person responsible for executing the test and recording the test results.

[Class A, B, C]

### 5.8 Software release

*Replace the existing title with the following:*

### 5.8 * Software RELEASE for utilization at a SYSTEM level

#### 5.8.1 Ensure software VERIFICATION is complete

*Replace the existing text with the following:*

The MANUFACTURER shall ensure that all software VERIFICATION ACTIVITIES have been completed and the results have been EVALUATED before the software is released. [Class A, B, C]

#### 5.8.2 Document known residual ANOMALIES

*Replace "*[Class B, C]*" with "*[Class A, B, C]*".*

#### 5.8.4 Document released VERSIONS

*Replace, in the existing text, "*SOFTWARE PRODUCT*" with "*MEDICAL DEVICE SOFTWARE*".*

#### 5.8.6 Ensure ACTIVITIES and TASKS are complete

*Replace the existing text with the following:*

The MANUFACTURER shall ensure that all software development plan (or maintenance plan) ACTIVITIES and TASKS are complete along with the associated documentation. [Class B, C]

NOTE See 5.1.3.b).

#### 5.8.7 Archive software

*Replace, in item a) "*SOFTWARE PRODUCT*" with "*MEDICAL DEVICE SOFTWARE*" and, in the second part of the sentence, "*device*" with "*MEDICAL DEVICE SOFTWARE*".*

*Replace "*[Class B, C]*" with "*[Class A, B, C]*".*

#### 5.8.8 Assure repeatability of software release

*Replace the existing title of this subclause with the following:*

### 5.8.8 Assure reliable delivery of released software

*Replace twice in the existing text the term "*software product*" by "*MEDICAL DEVICE SOFTWARE*".*

*Replace "*[Class B, C]*" with "*[Class A, B, C]*".*

## 6 SOFTWARE MAINTENANCE product

### 6.1 * Establish software maintenance plan

*Replace, in existing list item e), "*SYSTEM*" with "*SOFTWARE SYSTEM*".*

### 6.2 * Problem and modification analysis

### 6.2.1 Document and EVALUATE feedback

#### 6.2.1.1 Monitor feedback

*Replace the existing text with the following":*

The MANUFACTURER shall monitor feedback on MEDICAL DEVICE SOFTWARE released for intended use. [Class A, B, C]

#### 6.2.1.2 Document and EVALUATE feedback

*Replace, in the first sentence,* "SOFTWARE PRODUCT" *with* "MEDICAL DEVICE SOFTWARE".

#### 6.2.1.3 EVALUATE PROBLEM REPORT'S affects on SAFETY

*Replace the existing text with the following:*

Each PROBLEM REPORT shall be EVALUATED to determine how it affects the SAFETY of MEDICAL DEVICE SOFTWARE released for intended use (see 9.2) and whether a change to that software is needed to address the problem. [Class A, B, C]

### 6.2.2 Use software problem resolution PROCESS

*Replace the existing text of the note with the following:*

NOTE A problem could show that a SOFTWARE SYSTEM or SOFTWARE ITEM has not been placed in the correct software safety class. The problem resolution process can suggest changes of the software safety class. When the PROCESS has been completed, any change of safety class in the SOFTWARE SYSTEM or its SOFTWARE ITEMS should be made known and documented.

### 6.2.3 Analyse CHANGE REQUESTS

*Replace the existing text with the following:*

In addition to the analysis required by Clause 9, the MANUFACTURER shall analyse each CHANGE REQUEST for its effect on the organization, MEDICAL DEVICE SOFTWARE released for intended use, and SYSTEMS with which it interfaces. [Class A, B, C]

### 6.2.4 CHANGE REQUEST approval

*Replace* "SOFTWARE PRODUCTS" *with* "MEDICAL DEVICE SOFTWARE".

### 6.2.5 Communicate to users and regulators

*Replace* "SOFTWARE PRODUCTS" *with* "MEDICAL DEVICE SOFTWARE" (3 times).

### 6.3 * Modification implementation

### 6.3.1 Use established PROCESS to implement modification

*Replace the existing text with the following:*

The MANUFACTURER shall identify and perform any Clause 5 ACTIVITIES that need to be repeated as a result of the modification. [Class A, B, C]

NOTE For requirements relating to RISK MANAGEMENT of software changes see 7.4.

### 6.3.2 Re-release modified SOFTWARE SYSTEM

*Replace existing text with:*

The MANUFACTURER shall release modifications according to 5.8. [Class A, B, C]

NOTE Modifications can be released as part of a full re-release of a SOFTWARE SYSTEM or as a modification kit comprising changed SOFTWARE ITEMS and the necessary tools to install the changes as modifications to an existing SOFTWARE SYSTEM.

## 7 Software RISK MANAGEMENT PROCESS

### 7.1.5 Document sequences of events

*Delete this subclause.*

### 7.2 RISK CONTROL measures

### 7.2.1 Define RISK CONTROL measures

*Replace the existing text with the following:*

For each case documented in the RISK MANAGEMENT FILE where a SOFTWARE ITEM could contribute to a HAZARDOUS SITUATION, the MANUFACTURER shall define and document RISK CONTROL measures in accordance with ISO 14971.

[Class B, C]

NOTE The RISK CONTROL measures can be implemented in hardware, software, the working environment or user instruction.

### 7.2.2 RISK CONTROL measures implemented in software

*Replace the existing text of list item b) with the following:*

b) assign to each SOFTWARE ITEM that contributes to the implementation of a RISK CONTROL measure a software safety class based on the RISK that the RISK CONTROL measure is controlling (see 4.3 a)); and

### 7.3 Verification of risk control measures

### 7.3.1 Verify RISK CONTROL measures

*Add the following text after the first sentence and before existing "* [Class B, C]*":*

The MANUFACTURER shall review the RISK CONTROL measure and determine if it could result in a new HAZARDOUS SITUATION.

### 7.3.2 Document any new sequences of events

*Delete the existing title and text and add "Not used".*

## 8 * Software configuration management PROCESS

### 8.1 *Configuration identification

### 8.1.1 Establish means to identify CONFIGURATION ITEMS

*Replace the existing text with the following:*

The MANUFACTURER shall establish a scheme for the unique identification of CONFIGURATION ITEMS and their VERSIONS to be controlled according to the development and configuration planning specified in 5.1. [Class A, B, C]

### 8.1.2 Identify SOUP

*Delete, following list item c), the phrase* "of each SOUP CONFIGURATION ITEM being used."

### 8.2 * Change control

#### 8.2.1 Approve CHANGE REQUESTS

*Add, after the term "*CONFIGURATION ITEMS*" the following phrase: "*identified to be controlled according to 8.1".

*Replace, in Note 2, "*see 5.1.1 e)*" with "*see 5.1.1 d)*".*

#### 8.2.4 Provide means for TRACEABILITY of change

*Replace the existing text with the following:*

The MANUFACTURER shall maintain records of the relationships and dependencies between:

a) CHANGE REQUEST;

b) relevant PROBLEM REPORT; and

c) approval of the CHANGE REQUEST.

[Class A, B, C]

## 9 Software problem resolution PROCESS

### 9.1 Prepare PROBLEM REPORTS

*Replace the existing text with the following:*

The MANUFACTURER shall prepare a PROBLEM REPORT for each problem detected in the MEDICAL DEVICE SOFTWARE. PROBLEM REPORTS shall include a statement of criticality (for example, effect on performance, SAFETY, or SECURITY) as well as other information that may aid in the resolution of the problem (for example, devices affected, supported accessories affected).

[Class A, B, C]

NOTE Problems can be discovered before or after release, inside the MANUFACTURER'S organization or outside it.

### 9.5 Maintain records

*Delete, at the end of the second paragraph,* "(see 7.4)".

### 9.7 Verify software problem resolution

*Replace, in list item c), "*SOFTWARE PRODUCTS*" with "*MEDICAL DEVICE SOFTWARE*".*

## Annex A – Rationale for the requirements of this standard

## A.1 Rationale

*Replace, in the first sentence of the fourth paragraph, the term "hazardous situation" with the same term in small capitals: "*HAZARDOUS SITUATION*".*

Replace, in the second sentence of the fourth paragraph, the term *"*HAZARDS*" with "*HAZARDOUS SITUATIONS*".*

*Delete, in the second sentence of the final paragraph, the phrase: "*that cannot by definition cause a HAZARD*".*

*Delete, in the third sentence of the final paragraph, the word "easily".*

**Table A.1 – Summary of requirements by software safety class**

*Replace the existing table with the following:*

**Table A.1 – Summary of requirements by software safety class**

| Clauses and subclauses | | Class A | Class B | Class C |
|---|---|---|---|---|
| Clause 4 | All requirements | X | X | X |
| 5.1 | 5.1.1, 5.1.2, 5.1.3, 5.1.6, 5.1.7, 5.1.8, 5.1.9, | X | X | X |
| | 5.1.5, 5.1.10, 5.1.11, 5.1.12 | | X | X |
| | 5.1.4 | | | X |
| 5.2 | 5.2.1, 5.2.2, 5.2.4, 5.2.5, 5.2.6 | X | X | X |
| | 5.2.3 | | X | X |
| 5.3 | 5.3.1, 5.3.2, 5.3.3, 5.3.4, 5.3.6 | | X | X |
| | 5.3.5 | | | X |
| 5.4 | 5.4.1 | | X | X |
| | 5.4.2, 5.4.3, 5.4.4 | | | X |
| 5.5 | 5.5.1 | X | X | X |
| | 5.5.2, 5.5.3, 5.5.5 | | X | X |
| | 5.5.4 | | | X |
| 5.6 | All requirements | | X | X |
| 5.7 | All requirements | X | X | X |
| 5.8 | 5.8.1, 5.8.2, 5.8.4, 5.8.7, 5.8.8 | X | X | X |
| | 5.8.3, 5.8.5, 5.8.6 | | X | X |
| Clause 6 | All requirements | X | X | X |
| 7.1 | All requirements | | X | X |
| 7.2 | All requirements | | X | X |
| 7.3 | All requirements | | X | X |
| 7.4 | 7.4.1 | X | X | X |
| | 7.4.2, 7.4.3 | | X | X |
| Clause 8 | All requirements | X | X | X |
| Clause 9 | All requirements | X | X | X |



## Annex B – Guidance on the provisions of this standard

### B.1.1 Purpose

*Replace, in the second sentence of the first paragraph, the term "*SOFTWARE PRODUCTS*" by "*MEDICAL DEVICE SOFTWARE*".*

### B.1.2 Field of application

*Replace, in the final sentence of the second paragraph, "*MEDICAL DEVICE RISK MANAGEMENT PROCESS*" by "*the overall MEDICAL DEVICE RISK MANAGEMENT PROCESS".

*Replace, in the second sentence of the third paragraph, the term "*SOFTWARE PRODUCT(S)*" by "*MEDICAL DEVICE SOFTWARE*".*

## B.3 Terms and definitions

*Add, at the end of the second sentence of the second paragraph, following the words "*in its own right*" the phrase ",* which then becomes a software MEDICAL DEVICE*".*

## B.4 General requirements

### B.4.2 RISK MANAGEMENT

*Replace, in the last sentence of the second paragraph, the term "*HAZARDS*" WITH "*HAZARDOUS SITUATIONS*".*

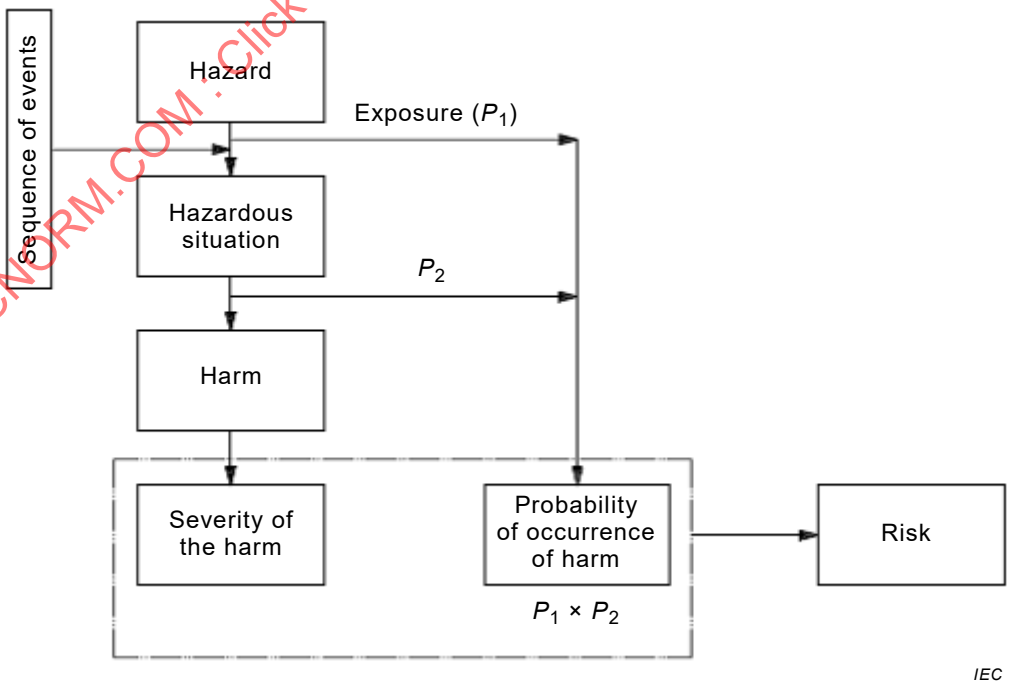
### B.4.3 Software safety classification

*Replace the existing second paragraph with the following:*

RISK is considered to be a combination of the severity of HARM and the probability of its occurrence. However, no consensus exists for a method of quantitatively estimating the probability of occurrence of a software failure. When software is present in a sequence or combination of events leading to a HAZARDOUS SITUATION, the probability of the software failure occurring cannot be considered in estimating the RISK for the HAZARDOUS SITUATION. In such cases, considering a worst case probability is appropriate, and the probability for the software failure occurring should be set to 1. When it is possible to estimate the probability for the remaining events in the sequence (as it may be if they are not software) that probability can be used for the probability of the HAZARDOUS SITUATION occurring ($P_1$ in Figure B.2).

In many cases however, it might not be possible to estimate the probability for the remaining events in the sequence, and the RISK should be EVALUATED on the basis of the nature of the HARM alone (the probability of the HAZARDOUS SITUATION occurring should be set to 1). RISK ESTIMATION in these cases should be focused on the SEVERITY of the HARM resulting from the HAZARDOUS SITUATION. Subjective rankings of probability can also be assigned based on clinical knowledge to distinguish failures that a clinician would be likely to detect from those that would not be detected and would be more likely to cause HARM.

Estimates of probability of a HAZARDOUS SITUATION leading to HARM ($P_2$ in Figure B.2) generally require clinical knowledge to distinguish between HAZARDOUS SITUATIONS where clinical practice would be likely to prevent HARM, and HAZARDOUS SITUATIONS that would be more likely to cause HARM.




NOTE $P_1$ is the probability of a hazardous situation occurring.

$P_2$ is the probability of a hazardous situation leading to harm.

**Figure B.2 – Pictorial representation of the relationship of HAZARD, sequence of events, HAZARDOUS SITUATION, and HARM – from ISO 14971:2007 Annex E**

*Add, at the end of the existing twelfth paragraph ("*The software ARCHITECTURE should promote…*"), the following two new sentences:*

Segregation is not restricted to physical (processor or memory partition) separation but includes any mechanism that prevents one SOFTWARE ITEM from negatively affecting another. The adequacy of a segregation is determined based on the RISKS involved and the rationale which is required to be documented.

*Replace, in the first sentence of the final paragraph, "*MEDICAL DEVICE software*" with "*MEDICAL DEVICE SOFTWARE*"* (the whole term in small capitals).

*Replace, in the third sentence of the final paragraph, the term "*HAZARDS*" with "*HAZARDOUS SITUATIONS *", twice.*

*Replace the final sentence of the final paragraph with the following:*

If segregation is not possible between SOFTWARE ITEMS X and Y, then SOFTWARE ITEM X must be classified in software safety class C.

*Add the following new subclause:*

### B.4.4 LEGACY SOFTWARE

Subclause 4.4 establishes a process for application of this standard to LEGACY SOFTWARE. Some geographies may require the MANUFACTURER to show conformity to the standard to obtain regulatory approval of the MEDICAL DEVICE SOFTWARE, even if that software was designed prior to the existence of the current version of the standard (LEGACY SOFTWARE). In this case, the requirements in 4.4 provide a method for the the MANUFACTURER to demonstrate compliance of LEGACY SOFTWARE to the standard.

A MANUFACTURER may determine that retrospective documentation of an already finished development-lifecycle performed as an isolated activity does not result in the reduction of RISK associated with the use of the product. The process results in the identification of a subset of ACTIVITIES defined in this standard which does result in reduction of RISK. Some additional goals implicit in the process are:

– required ACTIVITIES and resulting documentation should rely on and make use of, wherever possible, existing documentation, and
– a MANUFACTURER should utilize resources as effectively as possible to effect a reduction of RISK.

In addition to a plan identifying the subset of ACTIVITIES to execute, the process also results in objective evidence supporting safe continued use of the LEGACY SOFTWARE and a summary rationale for this conclusion.

The RISKS associated with the planned continued use of the LEGACY SOFTWARE depend on the context in which the LEGACY SOFTWARE will be used to create a SOFTWARE SYSTEM. The MANUFACTURER will document all identified MEDICAL DEVICE HAZARDS associated with the LEGACY SOFTWARE.

Subclause 4.4 requires a comprehensive assessment of available post-production field data obtained for the LEGACY SOFTWARE during the time it has been in production and use. Typical sources of post-production data include:

– adverse events attributable to the device,
– feedback received from users of the device, and
– ANOMALIES discovered by the MANUFACTURER.

Though no consensus exists for a method of prospectively estimating quantitatively the probability of occurrence of a software failure, such information may be available for LEGACY SOFTWARE, based on the usage of such software and EVALUATION of post-production data. If it is possible in such cases to quantitatively estimate the probability of events in the sequence, a quantitative value may be used for expressing the probability of the entire sequence of events occurring. If such quantitative estimation is not possible, considering a worst case probability is appropriate, and the probability for the software failure occurring should be assumed to be 1.

The MANUFACTURER determination of how the LEGACY SOFTWARE will be used in the overall MEDICAL DEVICE SYSTEM ARCHITECTURE is input to the assessment of RISK. The RISKS to be considered vary accordingly.

– When LEGACY SOFTWARE has been safely and reliably used and the MANUFACTURER wishes to continue use of the LEGACY SOFTWARE, the rationale for continued use rests primarily on the assessment of RISK based on post-production records.
– When LEGACY SOFTWARE is reused to create a new SOFTWARE SYSTEM, the intended use of the LEGACY SOFTWARE might be different from its original intended use. In this case the RISK assessment must take into account the modified set of HAZARDOUS SITUATIONS which can arise due to failures of the LEGACY SOFTWARE.
– A reused LEGACY SOFTWARE may be used for similar intended use but integrated into a new SOFTWARE SYSTEM. In this case the RISK assessment should take into account modification of architectural RISK CONTROL measures according to 5.3.

When LEGACY SOFTWARE will be changed and used within a new SOFTWARE SYSTEM, the MANUFACTURER should consider how the existing records of safe and reliable operation may be invalidated by the changes.

Changes to the LEGACY SOFTWARE should be performed according to Clauses 4 to 9 of this standard, including assessment of impact to RISK CONTROL measures according to 7.4. In the case of LEGACY SOFTWARE, existing RISK CONTROL measures may not be fully documented and special care should be taken to EVALUATE the potential impact of changes, utilizing available documented design records as well as expertise of individuals having knowledge of the system.

According to 4.4, the MANUFACTURER performs a gap analysis in order to determine the available documentation including objective evidence of performed TASKS done during development of the LEGACY SOFTWARE and compared to 5.2, 5.3, 5.7, and Clause 7. Typical steps to accomplish this gap analysis include

a) identification of the LEGACY SOFTWARE, including VERSION, revision and any other means, required for clear identification;
b) EVALUATION of existing DELIVERABLES corresponding to the deliverables required by 5.2, 5.3, 5.7, and Clause 7;
c) EVALUATION of available objective evidence, documenting the previously applied software development lifecycle model (as appropriate);
d) EVALUATION of the adequacy of existing RISK MANAGEMENT documentation, taking ISO 14971 into account.

Taking the performed gap analysis into account, the MANUFACTURER will EVALUATE the potential reduction in RISK resulting from the generation of the missing DELIVERABLES and associated ACTIVITIES, and create a plan to perform ACTIVITIES and generate DELIVERABLES to close these gaps.

Reduction of RISK should balance the benefit of applying the software development process according to Clause 5 against the possibility that modification of the LEGACY SOFTWARE without full knowledge of its development history could introduce new defects that increase the risk. Some of the elements of Clause 5 may be assessed to have little to no reduction of RISK when done after the fact. For example, detailed design and unit verification reduce RISK primarily during the process of developing new software or refactoring existing software. If these

objectives are not planned, performing the ACTIVITIES in isolation may create documentation but lead to no reduction in RISK.

At a minimum, the gap closure plan addresses missing SOFTWARE SYSTEM test records. If these do not exist or are not suitable to support a rationale to continue use of the LEGACY SOFTWARE, the gap closure plan should include creation of SOFTWARE SYSTEM requirements at a functional level according to 5.2 and tests according to 5.7.

The documented rationale for continued use of the LEGACY SOFTWARE builds on the available objective evidence and analysis obtained in the course of assessing the RISK and creating a gap closure plan appropriate for the context of LEGACY SOFTARE reuse.

The rationale makes a positive case for the safe and reliable performace of the LEGACY SOFTWARE in the planned reuse context, taking into account both the post-production records available for the LEGACY SOFTWARE and the RISK CONTROL MEASURES affected by filling process gaps.

After LEGACY SOFTWARE has been re-used according to 4.4, those parts of the LEGACY SOFTWARE for which gaps in DELIVERABLES remain, continue to be LEGACY SOFTWARE and may be considered for further re-use again according to 4.4. When gaps in deliverables are closed by changing the LEGACY SOFTWARE, the changes should be performed according to Clauses 4 to 9 of this standard.

## B.5 Software development PROCESS

### B.5.1 Software development planning

*Replace,in the third and fourth sentences of the second paragraph, the term "*MEDICAL DEVICE SOFTWARE PRODUCT*" with "*MEDICAL DEVICE SOFTWARE*".*

### B.5.3 Software ARCHITECTURAL design

*Replace the first sentence of the first paragraph with the following:*

This ACTIVITY requires the MANUFACTURER to define the major structural components of the software and identify their key responsibilities, their externally visible properties and the relationship among them.

*Add, at the end of the third sentence of the first paragraph, "*(see 5.3.5 and B.4.3)*".*

*Replace the first sentence of the second paragraph with the following:*

The software safety classification of SOFTWARE ITEMS during the software ARCHITECTURE ACTIVITY creates a basis for the subsequent choice of software PROCESSES.

### B.5.4 Software detailed design

*Replace, in the third sentence of the first paragraph, the words "*We have*" with "*This standard has*".*

*Replace the eighth and nineth sentences of the first paragraph with the following:*

It is necessary to define the design of the SOFTWARE UNITS and the interfaces in sufficient detail to permit its SAFETY and effectiveness to be objectively VERIFIED where this can be ensured using other requirements or design documentation.

*Add, at the end of the first paragraph, the following new sentence;*

Detailed design must also be concerned with the architecture of the MEDICAL DEVICE SOFTWARE.

*Add, after the fourth sentence of the third paragraph, the following new sentence:*

VERIFICATION of the design provides assurance that it implements the software ARCHITECTURE and is free from contradiction with the software ARCHITECTURE.

### B.5.6 Software integration and integration testing

*Replace, in the last sentence of the sixth paragraph, "*a SOFTWARE PRODUCT*" with "*MEDICAL DEVICE SOFTWARE.

### B.5.7 SOFTWARE SYSTEM testing

*Add, at the end of the fifth paragraph, "*(See B.6.3).".

*Replace, in the first sentence of the seventh parapraph (penultimate paragraph) the term "*HAZARD*" with "*RISK*".*

## B.6 Software maintenance PROCESS

### B.6.2 Problem and modification analysis

*Replace, in the fifth sentence of the first paragraph, "*HAZARD*" with "*HAZARDOUS SITUATION*" twice so that the sentence reads as follows:*

It is also important to verify that the modified software does not cause a HAZARDOUS SITUATION or mitigate a RISK in software that previously did not cause a HAZARDOUS SITUATION or mitigate RISKS.

*Replace, in the first sentence of the third paragraph, "*SOFTWARE PRODUCT*" with "*MEDICAL DEVICE SOFTWARE*".*

*Replace, in the second dashed item of the third paragraph the phrase "*SOFTWARE PRODUCTS are re-validated" with "MEDICAL DEVICE SOFTWARE is re-validated*".*

*Replace, in the third dashed item of the third paragraph, "*SOFTWARE PRODUCTS*" with "*MEDICAL DEVICE SOFTWARE*".*

### B.6.3 Modification implementation

*Insert, after the fourth sentence of the existing text, the following three new sentences:*

Regression analysis and testing are employed to provide assurance that a change has not created problems elsewhere in the MEDICAL DEVICE SOFTWARE. Regression analysis is the determination of the impact of a change based on review of the relevant documentation (e.g., software requirements specification, software design specification, source code, test plans, test cases, test scripts, etc.) in order to identify the necessary regression tests to be run. Regression testing is the rerunning of test cases that a program has previously executed correctly and comparing the current result to the previous result in order to detect unintended effects of a software change.

## Annex C – Relationship to other standards

### C.1 General

*Replace the existing Figure C.1 with the following new figure, in which references to IEC 62366-1 and IEC 82304-1 have been added to the medical device process standards and the medical device product standards respectively:*

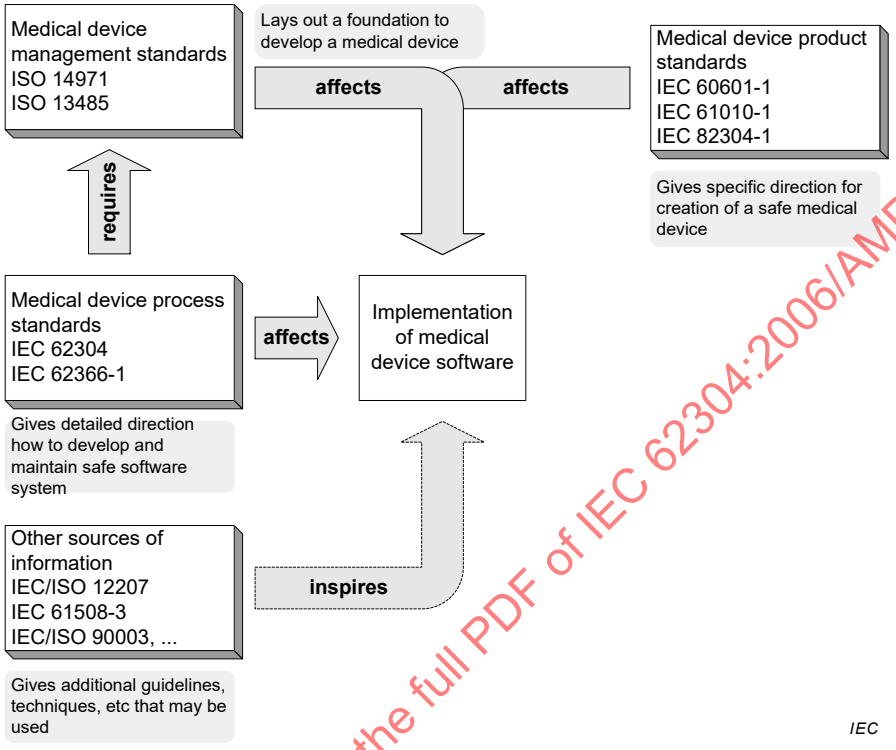



**Figure C.1 – Relationship of key MEDICAL DEVICE standards to IEC 62304**

### C.3 Relationship to ISO 14971

*Replace existing Table C.2 with the following:*

**Table C.2 – Relationship to ISO 14971:2007**

| ISO 14971:2007 clause | | Related clause of IEC 62304 | |
|---|---|---|---|
| 4.1 | RISK ANALYSIS process | | |
| 4.2 | Intended use and identification of characteristics related to the SAFETY of the MEDICAL DEVICE | | |
| 4.3 | Identification of HAZARDS | 7.1 | Analysis of software contributing to HAZARDOUS SITUATIONS |
| 4.4 | Estimation of the RISK(S) for each HAZARDOUS SITUATION | 4.3 | Software safety classification |
| 5 | RISK EVALUATION | | |
| 6.1 | RISK reduction | | |
| 6.2 | RISK CONTROL option analysis | 7.2.1 | Define RISK CONTROL measures |
| 6.3 | Implementation of RISK CONTROL measures | 7.2.2<br>7.3.1 | RISK CONTROL measures implemented in software<br>Verify RISK CONTROL measures |
| 6.4 | RESIDUAL RISK EVALUATION | | |
| 6.5 | RISK/benefit analysis | | |
| 6.6 | RISKS arising from RISK CONTROL MEASURES | 7.3.2 | Document any new sequences of events |
| 6.7 | Completeness of RISK CONTROL | | |
| 7 | Evaluation of overall RESIDUAL RISK acceptability | | |
| 8 | RISK MANAGEMENT report | 7.3.3 | Document TRACEABILITY |
| 9 | Production and post-production information | 7.4 | RISK MANAGEMENT of software changes |

## C.4 Relationship to PEMS requirements of IEC 60601-1:2005

*Replace the existing title of this clause with the following:*

## C.4 Relationship to PEMS requirements of IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012

### C.4.1 General

*Replace the existing text with the following:*

Requirements for software are a subset of the requirements for a programmable electrical medical system (PEMS). This standard identifies requirements for software which are in addition to, but not incompatible with, the requirements of IEC 60601-1:2005 + IEC 60601-1:2005 /AMD1:2012 [1] for PEMS. Because PEMS include elements that are not software, not all of the requirements of IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012 for PEMS are addressed in this standard. With the publication of IEC 60601-1:2005 + IEC 60601-1:2005 /AMD1:2012, IEC 62304 is now a normative reference of IEC 60601-1 and compliance with Clause 14 of IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012 (and thus compliance with the standard) requires compliance with parts of IEC 62304 (not with the whole of IEC 62304 because IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012 does not require compliance with post-production and maintenance requirements of IEC 62304). Finally, it is important to remember that IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012 is only used if the software is part of a PEMS and not if the software is itself a MEDICAL DEVICE.

### C.4.6 Coverage of PEMS requirements in IEC 60601-1

*Replace the existing title of this subclause with the following:*

### C.4.6 Coverage of PEMS requirements in IEC 60601-1:2005 + IEC 60601-1:2005 /AMD1:2012"

*Replace the existing Table C.3 with the following:*

**Table C.3 – Relationship to IEC 60601-1** *(1 of 5)*

| PEMS requirements from IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012 | Requirements of IEC 62304 relating to the software subsystem of a PEMS |
|---|---|
| **14.1 General**<br>The requirements in 14.2 to 14.12 (inclusive) shall apply to PEMS unless:<br>– none of the PROGRAMMABLE ELECTRONIC SUBSYSTEMS (PESS) provides functionality necessary for BASIC SAFETY or ESSENTIAL PERFORMANCE; or<br>– the application of RISK MANAGEMENTas described in 4.2 demonstrates that the failure of any PESS does not lead to an unacceptable RISK.<br>The requirements in 14.13 are applicable to any PEMS intended to be incorporated into an IT-NETWORK whether or not the requirements in 14.2 to 14.12 apply.<br>When the requirements in 14.2 to 14.13 apply, the requirements in subclause 4.3, Clause 5, Clause 7, Clause 8 and Clause 9 of IEC 62304:2006 shall also apply to the development or modification of software for each PESS. | **4.3 Software safety classification**<br>The PEMS requirements of IEC 60601-1 would only apply to software safety classes B and C. This standard includes some requirements for software safety class A.<br>The software development PROCESS required for compliance with IEC 60601-1 does not include the post production monitoring and maintenance required by Clause 6 of IEC 62304:2006. |
| **14.2 Documentation**<br>The documents required by Clause 14 shall be reviewed, approved, issued and changed in accordance with a formal document control procedure. | **5.1 Software development planning**<br>In addition to the specific requirements in the software development planning ACTIVITY, documents that are part of the RISK MANAGEMENT FILE are required to be maintained by ISO 14971. In addition, for documents that are required by the quality system, ISO 13485 [8] requires control of the documents. |
| **14.3 RISK MANAGEMENT PLAN**<br>The RISK MANAGEMENT plan required by 4.2.2 shall also include a reference to the PEMS VALIDATION plan (see 14.11). | Not specifically required.<br>There is no specific software validation plan. The PEMS validation plan is at the SYSTEM level and thus is outside the scope of this software standard. This standard does require TRACEABILITY from HAZARD to specific software cause to RISK CONTROL measure to VERIFICATION of the RISK CONTROL measure (see 7.3) |
| **14.4 PEMS DEVELOPMENT LIFE-CYCLE**<br>A PEMS DEVELOPMENT LIFE-CYCLE shall be documented. | **5.1 Software development planning**<br>**5.1.1 Software development plan**<br>The items addressed by the software development plan constitute a SOFTWARE DEVELOPMENT LIFE CYCLE. |
| The PEMS DEVELOPMENT LIFE-CYCLE shall contain a set of defined milestones. | |
| At each milestone, the ACTIVITIES to be completed and the VERIFICATION methods to be applied to those ACTIVITIES shall be defined. | **5.1.6 Software VERIFICATION planning**<br>VERIFICATION TASKS, milestones and acceptance criteria must be planned. |
| Each ACTIVITY shall be defined including its inputs and outputs. | **5.1.1 Software development plan**<br>ACTIVITIES are defined in this standard. Documentation to be produced is defined in each ACTIVITY. |

**Table C.3** *(2 of 5)*

| PEMS requirements from IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012 | Requirements of IEC 62304 relating to the software subsystem of a PEMS |
|---|---|
| Each milestone shall identify the RISK MANAGEMENT ACTIVITIES that must be completed before that milestone. | |
| The PEMS DEVELOPMENT LIFE-CYCLE shall be tailored for a specific development by making plans which detail ACTIVITIES, milestones and schedules. | **5.1.1 Software development plan**<br>This standard allows the development life cycle to be documented in the development plan. This means the development plan contains a tailored development life cycle. |
| The PEMS DEVELOPMENT LIFE-CYCLE shall include documentation requirements. | **5.1.1 Software development plan**<br>**5.1.8 Documentation planning** |
| **14.5 Problem resolution**<br>Where appropriate, a documented system for problem resolution within and between all phases and ACTIVITIES of the PEMS DEVELOPMENT LIFE-CYCLE shall be developed and maintained. | **9 Software problem resolution PROCESS** |
| Depending on the type of product, the problem resolution SYSTEM may:<br>– be documented as a part of the PEMS DEVELOPMENT LIFE-CYCLE;<br>– allow the reporting of potential or existing problems affecting BASIC SAFETY or ESSENTIAL PERFORMANCE;<br>– include an assessment of each problem for associated RISKS;<br>– identify the criteria that must be met for the issue to be closed;<br>– identify the action to be taken to resolve each problem. | **5.1.1 Software development plan**<br>**9.1 Prepare PROBLEM REPORTS** |
| **14.6 RISK MANAGEMENT PROCESS** | **7 Software RISK MANAGEMENT PROCESS** |
| **14.6.1 Identification of known and foreseeable HAZARDS**<br>When compiling the list of known or foreseeable HAZARDS, the MANUFACTURER shall consider those HAZARDS associated with software and hardware aspects of the PEMS including those associated with the incorporation of the PEMS into an IT-NETWORK, components of third-party origin and legacy subsystems. | **7.1 Analysis of software contributing to HAZARDOUS SITUATIONS**<br>This standard does not mention network/data coupling specifically |
| **14.6.2 RISK CONTROL**<br>Suitably validated tools and PROCEDURES shall be selected and identified to implement each RISK CONTROL measure. These tools and PROCEDURES shall be appropriate to assure that each RISK CONTROL measure satisfactorily reduces the identified RISK(S). | **5.1.4 Software development standards, methods and tools planning**<br>This standard requires the identification of specific tools and methods to be used for development in general, not for each RISK CONTROL measure. |
| **14.7 Requirements specification**<br>For the PEMS and each of its subsystems (e.g. for a PESS) there shall be a documented requirement specification. | **5.2 Software requirements analysis**<br>This standard deals only with the software subsystems of a PEMS. |
| The requirement specification for a system or subsystem shall include and distinguish any ESSENTIAL PERFORMANCE and any RISK CONTROL measures implemented by that system or subsystem. | **5.2.1** Define and document software requirements from SYSTEM requirements.<br>**5.2.2** Software requirements content<br>**5.2.3** Include RISK CONTROL measures in software requirements<br>This standard does not require that the requirements related to essential performance and RISK CONTROL measures be distinguished from other requirements, but it does require that all requirements be uniquely identified. |

**Table C.3** *(3 of 5)*

| PEMS requirements from IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012 | Requirements of IEC 62304 relating to the software subsystem of a PEMS |
|---|---|
| **14.8 ARCHITECTURE**<br>For the PEMS and each of its subsystems, an ARCHITECTURE shall be specified that shall satisfy the requirements specification. | **5.3 Software ARCHITECTURAL design** |
| Where appropriate, to reduce the RISK to an acceptable level, the ARCHITECTURE specification shall make use of:<br>a) COMPONENTS WITH HIGH-INTEGRITY CHARACTERISTICS;<br>b) fail-safe functions;<br>c) redundancy;<br>d) diversity;<br>e) partitioning of functionality;<br>f) defensive design, e.g. limits on potentially hazardous effects by restricting the available output power or by introducing means to limit the travel of actuators. | **5.3.5 Identify segregation necessary for RISK CONTROL**<br>Partitioning is the only technique identified, and it is only identified because there is a requirement to state how the integrity of the partitioning is assured. |
| The ARCHITECTURE specification shall take into consideration:<br>a) allocation of RISK CONTROL measures to subsystems and components of the PEMS;<br>b) failure modes of components and their effects;<br>c) common cause failures;<br>d) systemic failures;<br>e) test interval duration and diagnostic coverage;<br>f) maintainability;<br>g) protection from reasonably foreseeable misuse;<br>h) the IT-NETWORK specification, if applicable. | This is not included in this standard. |
| **14.9 Design and implementation**<br>Where appropriate, the design shall be decomposed into subsystems, each having both a design and test specification. | **5.4 Software detailed design**<br>**5.4.2 Develop detailed design for each SOFTWARE UNIT**<br>This standard does not require a test specification for detailed design. |
| Descriptive data regarding the design environment shall be included in the documentation. | **5.4.2 Develop detailed design for each SOFTWARE UNIT** |
| **14.10 VERIFICATION**<br>VERIFICATION is required for all functions that implement BASIC SAFETY, ESSENTIAL PERFORMANCE or RISK CONTROL measures. | **5.1.6 Software VERIFICATION planning**<br>VERIFICATION is required for each ACTIVITY |
| A VERIFICATION plan shall be produced to show how these functions shall be verified. The plan shall include:<br>– at which milestone(s) VERIFICATION is to be performed on each function;<br>– the selection and documentation of VERIFICATION strategies, ACTIVITIES, techniques, and the appropriate level of independence of the personnel performing the VERIFICATION;<br>– the selection and utilization of VERIFICATION tools;<br>– coverage criteria for VERIFICATION. | **5.1.6 Software VERIFICATION planning**<br>Independence of personnel is not included in this standard. It is considered covered in ISO 13485. |
| The VERIFICATION shall be performed according to the VERIFICATION plan. The results of the VERIFICATION ACTIVITIES shall be documented. | VERIFICATION requirements are in most of the ACTIVITIES. |
| **14.11 PEMS VALIDATION**<br>A PEMS VALIDATION plan shall include the validation of BASIC SAFETY and ESSENTIAL PERFORMANCE. | This standard does not cover software validation. PEMS validation is a SYSTEM level ACTIVITY and is outside the scope of this standard. |
| Methods used for PEMS VALIDATION shall be documented | This standard does not cover software validation. PEMS validation is a SYSTEM level ACTIVITY and is outside the scope of this standard. |
| The PEMS VALIDATION shall be performed according to the PEMS VALIDATION plan. The results of the PEMS VALIDATION ACTIVITIES shall be documented. | This standard does not cover software validation. PEMS validation is a SYSTEM level ACTIVITY and is outside the scope of this standard. |

**Table C.3** *(4 of 5)*

| PEMS requirements from IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012 | Requirements of IEC 62304 relating to the software subsystem of a PEMS |
|---|---|
| The person having the overall responsibility for the PEMS VALIDATION shall be independent of the design team. The MANUFACTURER shall document the rationale for the level of independence. | This standard does not cover software validation. PEMS validation is a SYSTEM level ACTIVITY and is outside the scope of this standard. |
| No member of a design team shall be responsible for the PEMS VALIDATION of their own design. | This standard does not cover software validation. PEMS validation is a SYSTEM level ACTIVITY and is outside the scope of this standard. |
| All professional relationships of the members of the PEMS VALIDATION team with members of the design team shall be documented in the RISK MANAGEMENT FILE. | This standard does not cover software validation. PEMS validation is a SYSTEM level ACTIVITY and is outside the scope of this standard. |
| A reference to the methods and results of the PEMS VALIDATION shall be included in the RISK MANAGEMENT FILE. | This standard does not cover software validation. PEMS validation is a SYSTEM level ACTIVITY and is outside the scope of this standard. |
| **14.12 Modification**<br>If any or all of a design results from a modification of an earlier design then either all of this clause applies as if it were a new design or the continued validity of any previous design documentation shall be assessed under a documented modification/change PROCEDURE. | **6 Software maintenance PROCESS**<br>This standard takes the approach that software maintenance should be planned and that implementation of modifications should use the software development PROCESS or an established software maintenance PROCESS. |
| When software is modified, the requirements in subclause 4.3, Clause 5, Clause 7, Clause 8 and Clause 9 of IEC 62304:2006 shall also apply to the modification. | |

**Table C.3** *(5 of 5)*

| PEMS requirements from IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012 | Requirements of IEC 62304 relating to the software subsystem of a PEMS |
|---|---|
| **14.13 PEMS intended to be incorporated into an IT-NETWORK**<br>If the PEMS is intended to be incorporated into an IT-NETWORK that is not validated by the PEMS MANUFACTURER, the MANUFACTURER shall make available instructions for implementing such connection including the following:<br>a) the purpose of the PEMS'S connection to an IT-NETWORK;<br>b) the required characteristics of the IT-NETWORK incorporating the PEMS;<br>c) the required configuration of the IT-NETWORK incorporating the PEMS;<br>d) the technical specifications of the network connection of the PEMS including security specifications;<br>e) the intended information flow between the PEMS, the IT-NETWORK and other devices on the IT-NETWORK, and the intended routing through the IT-NETWORK; and<br>NOTE 1 This can include aspects of effectiveness and data and system security as related to BASIC SAFETY and ESSENTIAL PERFORMANCE (see also Clause H.6 and IEC 80001-1:2010).<br>f) a list of the HAZARDOUS SITUATIONS resulting from a failure of the IT-NETWORK to provide the characteristics required to meet the purpose of the PEMS connection to the IT-NETWORK.<br>In the ACCOMPANYING DOCUMENTS, the MANUFACTURER shall instruct the RESPONSIBLE ORGANIZATION that:<br>– connection of the PEMS to an IT-NETWORK that includes other equipment could result in previously unidentified RISKS to PATIENTS, OPERATORS or third parties;<br>– the RESPONSIBLE ORGANIZATION should identify, analyze, evaluate and control these RISKS;<br>NOTE 3 IEC 80001-1:2010 provides guidance for the RESPONSIBLE ORGANIZATION to address these risks.<br>– subsequent changes to the IT-NETWORK could introduce new RISKS and require additional analysis; and<br>– changes to the IT-NETWORK include:<br>• changes in the IT-NETWORK configuration;<br>• connection of additional items to the IT-NETWORK;<br>• disconnecting items from the IT-NETWORK;<br>• update of equipment connected to the IT-NETWORK; and<br>• upgrade of equipment connected to the IT-NETWORK. | Requirements for incorporation into an IT-network are not included in this standard. |



### C.4.7 Relationship to requirements in IEC 60601-1-4

*Replace the existing text of this subclause, including Table C.4, with the following:*

IEC 60601-1-4 has been withdrawn.

## C.5 Relationship to IEC 61010-1

*Replace, in the first paragraph, the bibliographical reference "*[4]*" with "*[5]*".*

*Replace, in the third sentence of the third paragraph, the term "*HAZARD" with "HAZARDOUS SITUATION*".*

*Replace the existing fourth paragraph with the following:*

IEC 61010-1:2010 has a general requirement for risk assessment in Clause 17, which is more streamlined than the full risk management requirements of ISO 14971. Applying IEC 61010-1 Clause 17 alone does not meet the required criteria for risk management of IEC 62304, which is based on full ISO 14971 risk management requirements. With this in mind, it is expected by this standard that when an IVD medical device has software-related risks, its risk management process is performed following ISO 14971 instead of only Clause 17 of IEC 61010-1. Compliance with Clause 17 of IEC 61010-1 will be achieved, as detailed in the Note to Clause 17 of IEC 61010-1:

NOTE One RISK assessment procedure is outlined in Annex J. Other RISK assessment procedures are contained in ISO 14971, SEMI S10-1296, IEC 61508, ISO 14121-1, and ANSI B11.TR3. Other established procedures which implement similar steps can also be used.

The flowchart in Figure C.3 shows the application of IEC 62304 with IEC 61010-1, Clause 17:

## C.6 Relationship to ISO/IEC 12207

**Table C.5 – Relationship to ISO/IEC 12207**

*Replace the existing title and content of this table with the following:*

**Table C.5 – Relationship to ISO/IEC 12207:2008** *(1 of 6)*

| ISO/IEC 62304 PROCESSES | | ISO/IEC 12207:2008 | |
|---|---|---|---|
| ACTIVITY | TASK | PROCESSES | ACTIVITY/TASK |
| 5 Software development PROCESS | | | |
| 5.1 Software development planning | | | |
| | 5.1.1 Software development plan | 7.1.1 Software Implementation | 7.1.1.3.1 Software implementation strategy<br>7.1.1.3.1.1<br>7.1.1.3.1.3<br>7.1.1.3.1.4<br>6.3.1.3.2 Project planning<br>6.3.1.3.2.1 |
| | 5.1.2 Keep software development plan updated | 6.3.2 Project Assesment and Control | 6.3.2.3.2 Project control<br>6.3.2.3.2.1 |
| | 5.1.3 Software development plan reference to SYSTEM design and development | 6.4.3 System Architectural Design<br>6.4.5 System Integration<br>7.2.5 Software Validation Process | 6.4.3.3.1 Establishing architecture<br>6.4.3.3.1.1<br>6.4.5.3.1 Integration<br>6.4.5.3.1.1<br>7.2.5.3.1 Process implementation<br>7.2.5.3.1.4 |
| | 5.1.4 Software development standards, methods and tools planning | 7.1.1 Software Implementation | 7.1.1.3.1 Software implementation strategy<br>7.1.1.3.1.3 |
| | 5.1.5 Software integration and integration testing planning | 7.1.6 Software Integration | 7.1.6.3.1 Software integration<br>7.1.6.3.1.1 |
| | 5.1.6 Software VERIFICATION planning | 7.2.4 Software Verification<br>7.1.5 Software Construction<br>7.1.6 Software Integration<br>7.1.7 Software Qualification Testing | 7.2.4.3.1 Process implementation<br>7.2.4.3.1.4<br>7.2.4.3.1.5<br>7.1.5.3.1 Software construtions<br>7.1.5.3.1.5<br>7.1.6.3.1 Software integration<br>7.1.6.3.1.5<br>7.1.7.3.1 Software qualification testing<br>7.1.7.3.1.3 |
| | 5.1.7 Software RISK MANAGEMENT planning | 6.3.4 Risk Management Process | |
| | 5.1.8 Documentation planning | 7.2.1 Software Documentation Management | 7.2.1.3.1 Process implementation<br>7.2.1.3.1.1 |

**Table C.5** *(2 of 6)*

| ISO/IEC 62304 PROCESSES | | ISO/IEC 12207:2008 | |
|---|---|---|---|
| ACTIVITY | TASK | PROCESSES | ACTIVITY/TASK |
| | 5.1.9 Software configuration management planning | 7.2.2 Software Configuration Management<br>7.2.8 Software Problem Resolution | 7.2.2.3.1 Process implementation 7.2.2.3.1.1<br>7.2.8.3.1 Process implementation 7.2.8.3.1.1 |
| | 5.1.10 Supporting items to be controlled | 6.2.2 Infrastructure Management<br>6.2.2 Infrastructure Management | 6.2.2.3.2 Establishment of the infrastructure 6.2.2.3.2.1<br>6.2.2.3.3 Maintenance of the infrastructure 6.2.2.3.3.1 |
| | 5.1.11 Software CONFIGURATION ITEM control before VERIFICATION | 7.2.2 Software Configuration Management | 7.2.2.3.2 Configuration identification 7.2.2.3.2.1 |
| 5.2 Software requirements analysis | | | |
| | 5.2.1 Define and document software requirements from SYSTEM requirements | 6.4.3 System Architectural Design | 6.4.3.3.1 Establishing architecture 6.4.3.3.1.1 |
| | 5.2.2 Software requirements content | 7.1.2 Software Requirements Analysis | 7.1.2.3.1 Software requirements analysis 7.1.2.3.1.1 |
| | 5.2.3 Include RISK CONTROL measures in software requirements | | |
| | 5.2.4 Re-EVALUATE MEDICAL DEVICE RISK ANALYSIS | None | None |
| | 5.2.5 Update SYSTEM requirements | 7.1.2 Software Requirements Analysis | 7.1.2.3.1 Software requirements analysis 7.1.2.3.1.1 a) & b) |
| | 5.2.6 Verify software requirements | 7.2.4 Software Verification | 7.2.4.3.2 Verification 7.2.4.3.2.1 |

**Table C.5** *(3 of 6)*

| ISO/IEC 62304 PROCESSES | | ISO/IEC 12207:2008 | |
|---|---|---|---|
| ACTIVITY | TASK | PROCESSES | ACTIVITY/TASK |
| 5.3 Software ARCHITECTURAL design | 5.3.1 Transform software requirements into an ARCHITECTURE | 7.1.3 Software Architectural Design | 7.1.3.3.1 Software architectural design 7.1.3.3.1.1 |
| | 5.3.2 Develop an ARCHITECTURE for the interfaces of SOFTWARE ITEMS | | 7.1.3.3.1 Software architectural design 7.1.3.3.1.2 |
| | 5.3.3 Specify functional and performance requirements of SOUP item | None | none |
| | 5.3.4 Specify SYSTEM hardware and software required by SOUP item | None | none |
| | 5.3.5 Identify segregation necessary for RISK CONTROL | None | none |
| | 5.3.6 Verify software ARCHITECTURE | 7.1.3 Software Architectural Design | 7.1.3.3.1 Software architectural design 7.1.3.3.1.6 |
| 5.4 Software detailed design | 5.4.1 Refine SOFTWARE ARCHITECTURE into SOFTWARE UNITS | 7.1.4 Software Detailed Design | 7.1.4.3.1 Software detailed design 7.1.4.3.1.1 |
| | 5.4.2 Develop detailed design for each SOFTWARE UNIT | | |
| | 5.4.3 Develop detailed design for interfaces | | 7.1.4.3.1 Software detailed design 7.1.4.3.1.2 |
| | 5.4.4 Verify detailed design | 7.1.4 Software Detailed Design | 7.1.4.3.1 Software detailed design 7.1.4.3.1.7 |
| 5.5 SOFTWARE UNIT implementation and verification | 5.5.1 Implement each SOFTWARE UNIT | 7.1.5 Software Construction | 7.1.5.3.1 Software construction 7.1.5.3.1.1 |
| | 5.5.2 Establish SOFTWARE UNIT VERIFICATION PROCESS | 7.1.4 Software Detailed Design<br>7.1.5 Software Construction | 7.1.4.3.1 Software detailed design 7.1.4.3.1.5<br>7.1.5.3.1 Software Construction 7.1.5.3.1.5 |
| | 5.5.3 SOFTWARE UNIT acceptance criteria | 7.1.5 Software Construction | 7.1.5.3.1 Software construction 7.1.5.3.1.5 |
| | 5.5.4 Additional SOFTWARE UNIT acceptance criteria | 7.1.5 Software Construction<br>7.2.4 Software Verification | 7.1.5.3.1 Software construction 7.1.5.3.1.2 |
| | 5.5.5 SOFTWARE UNIT VERIFICATION | 7.1.5 Software Construction | 7.1.5.3.1 Software construction 7.1.5.3.1.2 |

**Table C.5** *(4 of 6)*

| ISO/IEC 62304 PROCESSES | | ISO/IEC 12207:2008 | |
|---|---|---|---|
| ACTIVITY | TASK | PROCESSES | ACTIVITY/TASK |
| 5.6 Software integration and integration testing | 5.6.1 Integrate SOFTWARE UNITS | 7.1.6 Software Integration | 7.1.6.3.1 Software integration<br>7.1.6.3.1.2 |
| | 5.6.2 Verify software integration | 7.1.6 Software Integration<br>6.4.5 System Integration | 7.1.6.3.1 Software integration<br>7.1.6.3.1.2<br>6.4.5.3.1 Integration<br>6.4.5.3.1.2 |
| | 5.6.3 Test integrated software | 7.1.7 Software Qualification Testing | 7.1.7.3.1 Software qualification testing<br>7.1.7.3.1.1 |
| | 5.6.4 Integration testing content | 7.1.7 Software Qualification Testing | 7.1.7.3.1 Software qualification testing<br>7.1.7.3.1.3 |
| | 5.6.5 Verify integration tests procedures | None | None |
| | 5.6.6 Conduct regression tests | 7.1.6 Software Integration | 7.1.6.3.1 Software integration<br>7.1.6.3.1.2 |
| | 5.6.7 Integration test record contents | 7.1.6 Software Integration | 7.1.6.3.1 Software integration<br>7.1.6.3.1.2 |
| | 5.6.8 Use software problem resolution PROCESS | 7.2.4 Software Verification | 7.2.4.3.1 Process implementation<br>7.2.4.3.1.6 |
| 5.7 SOFTWARE SYSTEM testing | 5.7.1 Establish tests for each software requirement | 7.1.6 Software Integration<br>7.1.7 Software Qualification Testing | 7.1.6.3.1 Software integration<br>7.1.6.3.1.4<br>7.1.7.3.1 Software qualification testing<br>7.1.7.3.1.1 |
| | 5.7.2 Use software problem resolution PROCESS | 7.2.4 Software Verification | 7.2.4.3.1 Process implementation<br>7.2.4.3.1.6 |
| | 5.7.3 Retest after changes | 7.2.8 Software Problem Resolution | 7.2.8.3.1 Process implementation<br>7.2.8.3.1.1 |
| | 5.7.4 Verify SOFTWARE SYSTEM testing | 7.1.7 Software Qualification Testing | 7.1.7.3.1 Software qualification testing<br>7.1.7.3.1.3 |
| | 5.7.5 SOFTWARE SYSTEM test record contents | 7.1.7 Software Qualification Testing | 7.1.7.3.1 Software qualification testing<br>7.1.7.3.1.1 |

**Table C.5** *(5 of 6)*

| ISO/IEC 62304 PROCESSES | | ISO/IEC 12207:2008 | |
|---|---|---|---|
| ACTIVITY | TASK | PROCESSES | ACTIVITY/TASK |
| 5.8 Software release | 5.8.1 Ensure software VERIFICATION is complete | 6.4.9 Software Operation<br>7.2.2 Software Configuration Management | 6.4.9.3.2 Operation activation and check-out<br>6.4.9.3.2.1<br>6.4.9.3.2.2<br>7.2.2.3.6 Release management and delivery<br>7.2.2.3.6.1 |
| | 5.8.2 Document known residual ANOMALIES | 7.2.2 Software Configuration Management<br>7.1.7 Software Qualification Testing | 7.2.2.3.5 Configuration evaluation<br>7.2.2.3.5.1<br>7.1.7.3.1 Software qualification testing<br>7.1.7.3.1.3 |
| | 5.8.3 EVALUATE known residual ANOMALIES | | |
| | 5.8.4 Document released VERSIONS | 7.2.2 Software Configuration Management Process | 7.2.2.3.6 Release management and delivery<br>7.2.2.3.6.1 |
| | 5.8.5 Document how released software was created | | |
| | 5.8.6 Ensure ACTIVITIES and TASKS are complete | | |
| | 5.8.7 Archive software | | |
| | 5.8.8 Assure repeatability of software release | | |
| 6 Software maintenance PROCESS | | 6.4.10 Software Maintenance Process | |
| 6.1 Establish software maintenance plan | | 6.4.10 Software Maintenance | None |
| 6.2 Problem and modification analysis | 6.2.1 Document and EVALUATE feedback | None | None |
| | 6.2.1.1 Monitor feedback | 6.4.10 Software Maintenance | None |
| | 6.2.1.2 Document and EVALUATE feedback | | |
| | 6.2.1.3 EVALUATE PROBLEM REPORT'S effects on SAFETY | 6.4.10 Software Maintenance | None |
| | 6.2.2 Use software problem resolution PROCESS | 6.4.10 Software Maintenance | None |
| | 6.2.3 Analyse CHANGE REQUESTS | 6.4.10 Software Maintenance | None |
| | 6.2.4 CHANGE REQUEST approval | 6.4.10 Software Maintenance | None |
| | 6.2.5 Communicate to users and regulators | 6.4.10 Software Maintenance | None |
| 6.3 Modification implementation | | None | None |
| | 6.3.1 Use established PROCESS to implement modification | 6.4.10 Software Maintenance | None |
| | 6.3.2 Re-release modified SOFTWARE SYSTEM | 7.2.2 Software Configuration Management | None |

**Table C.5** *(6 of 6)*

| ISO/IEC 62304 PROCESSES | | ISO/IEC 12207:2008 | |
|---|---|---|---|
| ACTIVITY | TASK | PROCESSES | ACTIVITY/TASK |
| 7 Software RISK MANAGEMENT PROCESS | | 6.3.4 Risk Management Process<br>This is based on ISO/IEC 16085. While there is some commonality it does not address the specific requirements for medical device software development with regard to risk management | |
| 8 Software configuration management PROCESS | | | |
| 8.1 Configuration identification | 8.1.1 Establish means to identify CONFIGURATION ITEMS | 7.2.2 Software Configuration Management | None |
| | 8.1.2 Identify SOUP | None | None |
| | 8.1.3 Identify SYSTEM configuration documentation | 7.2.2 Software Configuration Management | None |
| 8.2 Change control | 8.2.1 Approve CHANGE REQUESTS | 7.2.2 Software Configuration Management | None |
| | 8.2.2 Implement changes | 6.4.10 Software Maintenance | None |
| | 8.2.3 Verify changes | 7.2.2 Software Configuration Management | None |
| | 8.2.4 Provide means for TRACEABILITY of change | | |
| 8.3 Configuration status accounting | | 7.2.2 Software Configuration Management | None |
| 9 Software problem resolution PROCESS | | | |
| 9.1 Prepare PROBLEM REPORTS | | 7.2.8 Software Problem Resolution | None |
| 9.2 Investigate the problem | | 7.2.8 Software Problem Resolution | None |
| 9.3 Advise relevant parties | | 7.2.8 Software Problem Resolution | None |
| 9.4 Use change control process | | 7.2.2 Software Configuration Management<br>6.4.10 Software Maintenance | None |
| 9.5 Maintain records | | 7.2.8 Software Problem Resolution | None |
| 9.6 Analyse problems for trends | | 7.2.8 Software Problem Resolution | None |
| 9.7 Verify software problem resolution | | 7.2.8 Software Problem Resolution | None |
| 9.8 Test documentation contents | | All testing TASKS in ISO 12207 require documentation | None |



## C.7 Relationship to IEC 61508

*Insert, after the first sentence of the first paragraph the following two new sentences:*

The approach to safety in IEC 62304 is fundamentally different than the one in IEC 61508. IEC 62304 takes into account that the effectiveness of medical devices justifies residual risks related to their use.

*Replace the existing text of the fifth paragraph by the following:*

This standard simplifies issue 2) by defining the classification into 3 software safety classes based on the RISK caused by a failure. After classification, different PROCESSES are required for different software safety classes: the intention is to further reduce the probability (and/or the severity) of failure of the software.

## Bibliography

*Replace the existing references with the following:*

[1] IEC 60601-1:2005, *Medical electrical equipment – Part 1: General requirements for basic safety and essential performance*
IEC 60601-1:2005/AMD1:2012

[2] IEC 60601-1-4:1996, *Medical electrical equipment – Part 1: General requirements for safety – 4.Collateral standard: Programmable electrical medical systems* (withdrawn)
IEC 60601-1-4:1996/AMD1:1999

[3] IEC 60601-1-6, *Medical electrical equipment – Part 1-6: General requirements for basic safety and essential performance – Collateral standard: Usability*

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

[5] IEC 61010-1:2010 *Safety requirements for electrical equipment for measurement, control, and laboratory use – Part 1: General requirements*

[6] ISO 9000:2005, *Quality management systems – Fundamentals and vocabulary*

[7] ISO 9001:2008, *Quality management systems – Requirements*

[8] ISO 13485:2003, *Medical devices – Quality management systems – Requirements for regulatory purposes*

[9] ISO/IEC 12207:2008, *Systems and software engineering – Software life cycle processes*

[10] ISO/IEC 14764:1999, *Software Engineering – Software Life Cycle Processes – Maintenance*

[11] ISO/IEC 15504-5:2012, *Information technology – Process assessment – Part 5: An exemplar software life cycle process assessment model*

[12] ISO/IEC 25010:2011, *Systems and software engineering – System and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*

[13] ISO/IEC 33001:—[2)], *Information technology – Process assessment – Concepts and terminology*

[14] ISO/IEC 33004:—[2)], *Information technology – Process assessment – Requirements for process reference, process assessment and maturity modelss*

[15] ISO/IEC 90003:2014, *Software engineering – Guidelines for the application of ISO 9001:2008 to computer software*

[16] ISO/IEC Guide 51:2014, *Safety aspects – Guidelines for their inclusion in standards*

[17] IEEE 610.12:1990, *IEEE standard glossary of software engineering terminology*

[18] IEEE 1044:2009, *IEEE standard classification for software anomalies*



2 To be published.

[19] U.S. Department Of Health and Human Services, Food and Drug Administration, Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices, May 11, 2005, <://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm089543.htm>

[20] U.S. Department Of Health and Human Services, Food and Drug Administration, *General Principles of Software Validation; Final Guidance for Industry and FDA Staff*, January 11, 2002, <http://www.fda.gov/downloads/RegulatoryInformation/Guidances/ucm126955.pdf>

[21] IEC 62366-1:2015, *Medical devices – Part 1: Application of usability engineering to medical devices*

[22] IEC 82304-1:—[3], *Healthcare Software Systems – Part 1: General requirements*



3 In preparation.

# AVANT-PROPOS

Le présent amendement a été établi par un groupe de travail mixte du sous-comité 62A: Aspects généraux des équipements électriques utilisés en pratique médicale, du comité d'études 62 de l'IEC: Équipements électriques dans la pratique médicale, et du comité technique 210 de l'ISO, Management de la qualité et aspects généraux correspondants des DISPOSITIFS MEDICAUX.

La présente publication est publiée en tant que norme sous double logo.

La présente version bilingue (2017-09) correspond à la version anglaise monolingue publiée en 2015-06.

Le texte anglais de cet amendement est issu des documents 62A/1007/FDIS et 62A/1014/RVD.

Le rapport de vote 62A/1014/RVD donne toute information sur le vote ayant abouti à l'approbation de cet amendement.

La version française de cet amendement n'a pas été soumise au vote. Le comité a décidé que le contenu de cet amendement et de la publication de base ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "http://webstore.iec.ch" dans les données relatives à la publication recherchée. A cette date, la publication sera

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



# INTRODUCTION A L'AMENDEMENT 1

La première édition de l'IEC 62304 a été publiée en 2006. Le présent amendement vise à ajouter des exigences portant sur les LOGICIELS HERITES dont la conception du logiciel préexiste à la version actuelle, afin d'aider les fabricants qui doivent prouver la conformité à la norme pour satisfaire aux Directives européennes. Les modifications de la classification de sécurité du logiciel exigées pour le présent amendement comprennent la clarification des exigences et la mise à jour de la classification de sécurité du logiciel en vue d'inclure une approche fondée sur les risques. Les travaux se poursuivent en parallèle en vue de la deuxième édition de l'IEC 62304.

## AVANT-PROPOS

*Ajouter la note suivante à la fin de l'Avant-propos:*

NOTE L'attention des Comités Nationaux est attirée sur le fait que les FABRICANTS d'appareils et les organismes d'essai peuvent avoir besoin d'une période transitoire après la publication d'une nouvelle publication IEC ou ISO, ou d'une publication amendée ou révisée, pour fabriquer des produits conformes aux nouvelles exigences et pour adapter leurs équipements aux nouveaux essais ou aux essais révisés. Le comité recommande que le contenu de cette publication soit entériné au niveau national au plus tôt 3 ans après la date de publication.

## INTRODUCTION

*Remplacer, dans le deuxième alinéa, la troisième phrase existante par la suivante:*

Chaque PROCESSUS du cycle de vie comprend un ensemble d'ACTIVITES dont la plupart consistent en un ensemble de TACHES.

*Remplacer, dans la première phrase du quatrième alinéa, les mots «contribution d'un logiciel à un* DANGER*» par «contribution d'un logiciel à une* SITUATION DANGEREUSE*».*

*Remplacer, dans la deuxième phrase du quatrième alinéa, le terme* «DANGERS» *par* «SITUATIONS DANGEREUSES».

*Ajouter, après le sixième alinéa existant, le nouvel alinéa suivant:*

L'Amendement 1 met à jour la norme pour ajouter des exigences afin de traiter les LOGICIELS HERITES conçus avant la version actuelle afin d'aider les fabricants qui doivent prouver la conformité à la norme pour satisfaire aux Directives européennes et afin que les modifications de la classification de sécurité du logiciel comprennent la clarification des exigences et la mise à jour de la classification de sécurité du logiciel en vue d'inclure une approche fondée sur les risques.

## 1 Domaine d'application

### 1.2 * Domaine d'application

*Remplacer tout le texte existant de ce paragraphe par ce qui suit:*

La présente norme s'applique au développement et à la maintenance des LOGICIELS DE DISPOSITIFS MEDICAUX lorsque le logiciel est un DISPOSITIF MEDICAL en soi ou lorsque le logiciel est incorporé ou fait partie intégrante du DISPOSITIF MEDICAL final.

NOTE 1 La présente norme peut être utilisée dans le développement et la maintenance d'un logiciel qui est, en soi, un dispositif médical. Cependant, d'autres activités de développement sont nécessaires au niveau du système avant de pouvoir mettre ce type de logiciel en service. Ces activités au niveau du système ne sont pas couvertes par la présente norme, mais peuvent être consultées dans l'IEC 82304-1[1] [22].

La présente norme décrit les PROCESSUS qui sont conçus pour s'appliquer à un logiciel qui s'exécute sur un processeur, ou qui est exécuté par un autre logiciel (par exemple, un interpréteur) qui s'exécute sur un processeur.

La présente norme s'applique indépendamment du ou des dispositifs de stockage permanent utilisés pour stocker le logiciel (par exemple: disque dur, disque optique, mémoire permanente ou flash).

La présente norme s'applique indépendamment de la méthode de distribution du logiciel (par exemple: transmission par réseau ou par courrier électronique, sur disque optique, mémoire flash ou EEPROM). La méthode de distribution du logiciel proprement dite n'est pas considérée comme un LOGICIEL DE DISPOSITIF MEDICAL.

La présente norme ne couvre pas la validation et la mise sur le marché du DISPOSITIF MEDICAL, même lorsque le DISPOSITIF MEDICAL est intégralement constitué du logiciel.



---

1 En cours d'élaboration.

NOTE 2 Si un dispositif médical intègre un logiciel incorporé conçu pour être exécuté sur un processeur, les exigences de la présente norme s'appliquent au logiciel, y compris les exigences concernant les logiciels de provenance inconnue (voir 8.1.2).

NOTE 3 La validation et d'autres activités de développement sont nécessaires au niveau du système avant de pouvoir mettre en service le logiciel et le dispositif médical. Ces activités au niveau du système ne sont pas couvertes par la présente norme, mais elles peuvent être consultées dans les normes de produits associées (par exemple l'IEC 60601-1, l'IEC 82304-1, etc.).

### 1.4 Conformité

*Supprimer, dans le deuxième alinéa, l'instruction* «Voir Annexe D».

*Ajouter, après la Note 4 existante, la nouvelle note suivante:*

NOTE 5 Concernant la conformité des LOGICIELS HERITES, voir 4.4.

## 3 * Termes et définitions

**3.2**
**ANOMALIE**

*Remplacer, dans la définition,* «PRODUITS LOGICIELS» *par* «LOGICIELS DE DISPOSITIFS MEDICAUX».

Remplacer la *référence de* source existante par la *note suivante:*

NOTE Basée sur la définition 3.1 de l'IEEE 1044:1993,

**3.4**
**DEMANDE DE MODIFICATION**

*Remplacer* «PRODUIT LOGICIEL» *par* «LOGICIEL DE DISPOSITIF MEDICAL».

**3.5**
**ELEMENT DE CONFIGURATION**

*Remplacer, dans la note,* «ISO/IEC 12207:1995, définition 3.6» *par* «ISO/IEC 12207:2008, 4.7».

**3.7**
**EVALUATION**

*Remplacer la référence de source existante par* «[ISO/IEC 12207:2008, 4.12]».

**3.8**
**DOMMAGE**

*Remplacer la référence de source existante par* «[ISO 14971:2007, 2.2]».

**3.9**
**PHENOMENE DANGEREUX (DANGER)**

*Remplacer la référence de source existante par* «[ISO 14971:2007, 2.3]».

**3.10**
**FABRICANT**

*Ajouter les nouvelles notes suivantes:*

NOTE 1 L'attention est attirée sur le fait que les dispositions des règlements nationaux ou régionaux peuvent s'appliquer à la définition de fabricant.

NOTE 2 Pour une définition de l'étiquetage, voir la définition 3.6 de l'ISO 13485:2003.

*Remplacer la référence de source existante par* «[ISO 14971:2007, 2.8]».

**3.11**
**DISPOSITIF MEDICAL**

*Ajouter la nouvelle note suivante:*

NOTE 3 Conjointement aux normes IEC 60601-1:2005 et IEC 60601-1:2005/AMD1:2012, le terme «dispositif médical» a la même signification qu'APPAREIL EM ou SYSTEME EM (termes définis dans l'IEC 60601-1).

**3.12**
**LOGICIEL DE DISPOSITIF MEDICAL**

*Remplacer la définition existante par la suivante:*

SYSTEME LOGICIEL qui a été développé pour être incorporé dans le DISPOSITIF MEDICAL en cours de développement ou qui est destiné à être utilisé comme un DISPOSITIF MEDICAL

NOTE Ceci comprend un produit logiciel de DISPOSITIF MEDICAL qui est alors un DISPOSITIF MEDICAL à part entière.

**3.13**
**RAPPORT DE PROBLEME**

*Remplacer dans la définition et dans les Notes 1 et 2,* «PRODUIT LOGICIEL» *par* «LOGICIEL DE DISPOSITIF MEDICAL».

**3.16**
**RISQUE**

*Remplacer la référence de source existante par* «[ISO 14971:2007, 2.16]».

**3.17**
**ANALYSE DU RISQUE**

*Remplacer la référence de source existante par* «[ISO 14971:2007, 2.17]».

**3.18**
**MAITRISE DU RISQUE**

*Remplacer la référence de source existante par* «[ISO 14971:2007, 2.19]».

**3.19**
**GESTION DES RISQUES**

*Remplacer la référence de source existante par* «[ISO 14971:2007, 2.22, modifiée — L'expression «de contrôle» a été supprimée]».

**3.20**
**DOSSIER DE GESTION DES RISQUES**

*Remplacer la référence de source existante par* «[ISO 14971:2007, 2.23]».

**3.21**
**SECURITE**

*Remplacer la référence de source existante par* «[ISO 14971:2007, 2.24]».

**3.22**
**SURETE**

*Remplacer la définition existante par la suivante:*

protection des informations et des données de sorte que les personnes ou systèmes non autorisés ne puissent pas les lire ou les modifier et que leur accès aux personnes ou systèmes autorisés ne leur soit pas refusé.

NOTE Basée sur l'ISO/IEC 12207:2008, 4.39.

**3.23**
**BLESSURE GRAVE**

Supprimer, dans la première ligne *de la définition, les mots* «directement ou indirectement».

**3.24**
**MODELE DU CYCLE DE VIE DE DEVELOPPEMENT DU LOGICIEL**

*Supprimer, dans la deuxième ligne de la définition, les mots* «sa mise en fabrication».

*Remplacer, au premier tiret, les mots* «un PRODUIT LOGICIEL» *par* «LOGICIEL DE DISPOSITIF MEDICAL».

**3.25**
**ELEMENT LOGICIEL**

*Remplacer la définition existante par la suivante:*

toute partie identifiable d'un programme informatique, c'est-à-dire: code source, code objet, code de contrôle, données de contrôle, ou un ensemble de ces éléments

NOTE 1 Trois termes identifient la décomposition du logiciel. Le niveau supérieur est le SYSTEME LOGICIEL. Le niveau le plus bas qui n'est pas décomposé plus avant est L'UNITE LOGICIELLE. Tous les niveaux de composition, y compris les niveaux supérieur et inférieur, peuvent être dénommés ELEMENTS LOGICIELS. Un SYSTEME LOGICIEL est ainsi composé d'un ou de plusieurs ELEMENTS LOGICIELS et chaque ELEMENT LOGICIEL est composé d'une ou de plusieurs UNITES LOGICIELLES ou d'un ou de plusieurs ELEMENTS LOGICIELS décomposables. Il incombe au FABRICANT de fournir la définition et la granularité des ELEMENTS LOGICIELS et des UNITES LOGICIELLES.

NOTE 2 Basée sur l'ISO/IEC 90003:2004, 3.14 et ISO/IEC 12207:2008, 4.41.

**3.26**
**PRODUIT LOGICIEL**

*Supprimer le terme et la définition existants et ajouter* "Non utilisé".

**3.28**
**UNITE LOGICIELLE**

*Remplacer la note existante par la note suivante:*

NOTE La granularité des UNITES LOGICIELLES est définie par le FABRICANT (voir B.3).

**3.29**
**SOUP (sigle pour l'anglais «Software Of Unknown Provenance»)**
**logiciel de provenance inconnue**

*Remplacer, dans la troisième ligne de la définition,* «logiciel précédemment développé» *par* «ELEMENT LOGICIEL précédemment développé»

*Ajouter la nouvelle note suivante:*

NOTE Un SYSTEME LOGICIEL DE DISPOSITIF MEDICAL proprement dit ne peut pas être considéré comme un logiciel SOUP.

**3.30**
**SYSTEME**

*Remplacer la référence de source existante par la note suivante:*

NOTE Basée sur l'ISO/IEC 12207:2008, 4.48.

**3.32**
**TRAÇABILITE**

*Ajouter la nouvelle note suivante:*

NOTE Les exigences, l'architecture, les mesures de maîtrise du risque, etc. sont des exemples de livrables du PROCESSUS de développement.

**3.34**
**VERSION**

*Remplacer, dans le texte existant de la Note 1, les mots* «un PRODUIT LOGICIEL» *par* «LOGICIEL DE DISPOSITIF MEDICAL».

*Remplacer le texte existant de la Note 2 par ce qui suit*

NOTE 2 Basée sur la définition 4.56 de l'ISO/IEC 12207:2008.

*Ajouter les nouvelles définitions suivantes:*

**3.35**
**SITUATION DANGEREUSE**
situation dans laquelle des personnes, des biens ou l'environnement sont exposés à un ou à plusieurs PHENOMENES DANGEREUX

[SOURCE: ISO 14971:2007, 2.4]

**3.36**
**LOGICIEL HERITE**
LOGICIEL DE DISPOSITIF MEDICAL ayant été commercialisé légalement et étant toujours commercialisé actuellement, mais pour lequel les preuves tangibles de son développement conformément à la version actuelle de la présente norme sont insuffisantes

**3.37**
**DIFFUSION**
VERSION particulière d'un ELEMENT DE CONFIGURATION disponible à une fin particulière

NOTE Basée sur la définition 4.35 de l'ISO/IEC 12207:2008.

**3.38**
**RISQUE RESIDUEL**
RISQUE subsistant après que des mesures de MAITRISE DU RISQUE ont été prises

NOTE 1 Adaptée de l'ISO/IEC Guide 51:1999, définition 3.9.

NOTE 2 L'ISO/IEC Guide 51:1999, définition 3.9, utilise le terme «mesures de prévention» plutôt que «mesures de MAITRISE DU RISQUE». Toutefois, dans le cadre de la présente Norme internationale, les «mesures de prévention» ne sont qu'une option de maîtrise du RISQUE comme décrit en 6.2 [de l'ISO 14971:2007].

[SOURCE: ISO 14971:2007, 2.15].

**3.39**
**ESTIMATION DU RISQUE**
PROCESSUS utilisé pour attribuer des valeurs à la probabilité d'occurrence d'un DOMMAGE et à sa gravité

[SOURCE: ISO 14971:2007 2.20]

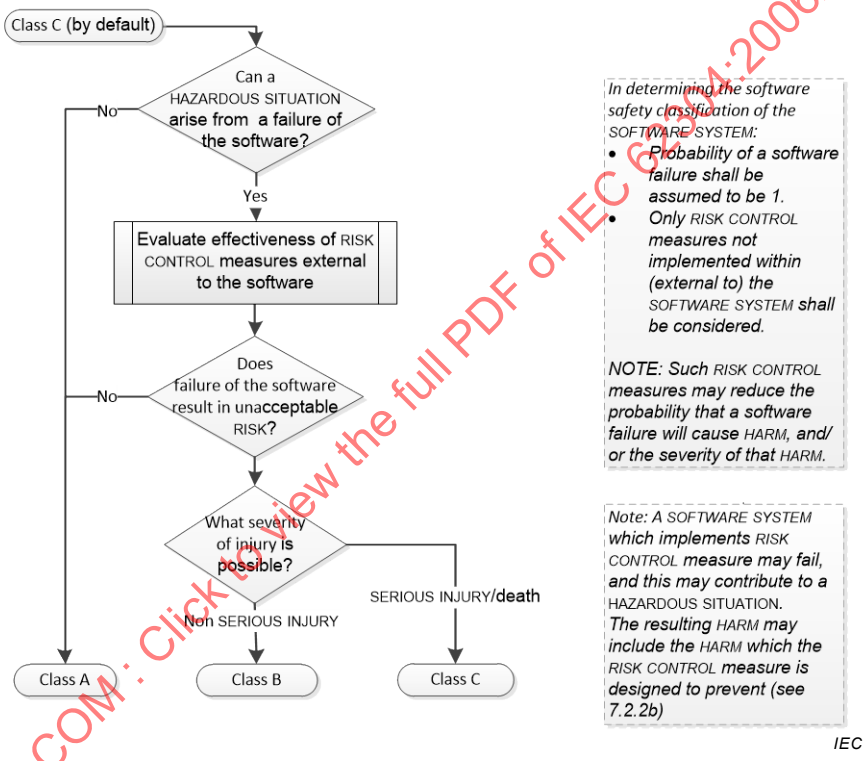
**3.40**
**EVALUATION DU RISQUE**
PROCESSUS de comparaison du RISQUE estimé avec les critères de RISQUE donnés afin de déterminer l'acceptabilité du RISQUE

[SOURCE: ISO 14971:2007 2.21]

### 4.3 * Classification de sécurité du logiciel

*Remplacer les points a) et b), et insérer une nouvelle Figure 3,comme suit:*

a) Le FABRICANT doit attribuer à chaque SYSTEME LOGICIEL une classe de sécurité du logiciel (A, B ou C) en fonction du RISQUE de DOMMAGE pour le patient, l'opérateur ou d'autres personnes résultant d'une SITUATION DANGEREUSE à laquelle le SYSTEME LOGICIEL peut contribuer dans un scénario représentant le cas le plus défavorable, tel qu'indiqué dans la Figure 3.




| Anglais | Français |
|---|---|
| Class C (by default) | Classe C (par défaut) |
| Can a HAZARDOUS SITUATION arise from a failure of the software? | Une SITUATION DANGEREUSE peut-elle se produire suite à une défaillance logicielle? |
| No | Non |
| Yes | Oui |
| Evaluate effectiveness of RISK CONTROL measures external to the software | Évaluer l'efficacité des mesures de MAITRISE DU RISQUE externes au logiciel |
| Does failure of the software result in unacceptable RISK? | La défaillance logicielle entraîne-t-elle un RISQUE inacceptable? |
| What severity of injury is possible? | Quelle gravité de blessure est possible? |
| SERIOUS INJURY/death | BLESSURE GRAVE/mort |
| Non SERIOUS INJURY | BLESSURE NON GRAVE |
| Class A | Classe A |

| Anglais | Français |
|---|---|
| Class B | Classe B |
| *In determining the software safety classification of the* SOFTWARE SYSTEM*:*<br>• *Probability of a software failure shall be assumed to be 1.*<br>• *Only* RISK CONTROL *measures not implemented within (external to) the* SOFTWARE SYSTEM *shall be considered.*<br>*NOTE: Such* RISK CONTROL *measures may reduce the probability that a software failure will cause HARM and/or the severity of that HARM.* | *Lors de la détermination de la classification de sécurité du logiciel du* SYSTEME LOGICIEL*:*<br>*. La probabilité d'une défaillance logicielle doit être évaluée à 1.*<br>*. Seules les mesures de* MAITRISE DU RISQUE *non mises en œuvre dans le (externe au)* SYSTEME LOGICIEL *doivent être considérées.*<br>*NOTE: Ces mesures de* MAITRISE DU RISQUE *peuvent réduire la probabilité qu'une défaillance logicielle provoque un* DOMMAGE *et/ou la gravité du* DOMMAGE*.* |
| *Note: A* SOFTWARE SYSTEM *which implements* RISK CONTROL *measure may fail, and this may contribute to a* HAZARDOUS SITUATION*. The resulting* HARM *may include the* HARM *which the* RISK CONTROL *measure is designed to prevent (see 7.2.2b)* | *Note: Un* SYSTEME LOGICIEL *qui met en œuvre des mesures de* MAITRISE DU RISQUE *peut connaître une défaillance, ce qui peut contribuer à une* SITUATION DANGEREUSE*. Le* DOMMAGE *qui en résulte peut inclure le* DOMMAGE *que la mesure de* MAITRISE DU RISQUE *est destinée à empêcher (voir 7.2.2b).* |

**Figure 3 – Affectation de la classification de sécurité du logiciel**

Le SYSTEME LOGICIEL est de classe A de sécurité du logiciel si:

– le SYSTEME LOGICIEL ne peut pas contribuer à une SITUATION DANGEREUSE; ou

– le SYSTEME LOGICIEL peut contribuer à une SITUATION DANGEREUSE qui ne résulte pas en un RISQUE inacceptable après étude des mesures de MAITRISE DU RISQUE externes au SYSTEME LOGICIEL.

Le SYSTEME LOGICIEL est de classe B de sécurité du logiciel si:

– le SYSTEME LOGICIEL peut contribuer à une SITUATION DANGEREUSE qui résulte en un RISQUE inacceptable après étude des mesures de MAITRISE DU RISQUE externes au SYSTEME LOGICIEL et que le DOMMAGE possible résultant est une BLESSURE NON GRAVE.

Le SYSTEME LOGICIEL est de classe C de sécurité du logiciel si:

– le SYSTEME LOGICIEL peut contribuer à une SITUATION DANGEREUSE qui résulte en un RISQUE inacceptable après étude des mesures de MAITRISE DU RISQUE externes au SYSTEME LOGICIEL et que le DOMMAGE possible résultant est la mort ou une BLESSURE GRAVE.

Pour un SYSTEME LOGICIEL initialement classifié comme présentant une classe de sécurité du logiciel B ou C, le FABRICANT peut mettre en œuvre d'autres mesures de MAITRISE DU RISQUE, externes au SYSTEME LOGICIEL (y compris la révision de l'architecture système contenant le SYSTEME LOGICIEL), et attribuer par la suite une nouvelle classification de sécurité du logiciel au SYSTEME LOGICIEL.

NOTE 1 Les mesures externes de MAITRISE DU RISQUE peuvent consister en du matériel, un SYSTEME LOGICIEL indépendant, des procédures de soins de santé ou d'autres moyens pour réduire le plus possible la contribution du logiciel aux SITUATIONS DANGEREUSES.

NOTE 2 Voir 3.2 de l'ISO 14971:2007, *Responsabilités de la direction*, pour la définition de l'acceptabilité du risque.

b) Non utilisé.

*Ajouter, à la fin de la première phrase du point d), les mots entre parenthèses suivants:* «(classes de sécurité du logiciel attribuées selon 4.3 a) remplaçant “SYSTEME LOGICIEL” par “ELEMENT LOGICIEL”)".

*Remplacer le texte existant du point f) par ce qui suit:*

f) Pour la conformité à la présente norme, lorsque la présente norme est appliquée à un groupe d'ELEMENTS LOGICIELS, le FABRICANT doit utiliser les PROCESSUS et TACHES qui sont

exigés par la classification de sécurité de l'ELEMENT LOGICIEL la plus élevée définie dans le groupe, à moins que le FABRICANT ne justifie par écrit dans le DOSSIER DE GESTION DES RISQUES l'utilisation d'une classification plus basse.

*Remplacer le texte existant de la note par ce qui suit:*

NOTE Dans les articles et les paragraphes qui suivent, les classes de sécurité du logiciel pour lesquelles une exigence spécifique s'applique sont identifiées selon l'exigence sous la forme de [Classe . . .].

*Ajouter le nouveau paragraphe suivant:*

### 4.4 * LOGICIEL HERITE

#### 4.4.1 Généralités

En variante à l'application des Articles 5 à 9 de la présente norme, la conformité du LOGICIEL HERITE peut être démontrée, comme indiqué du 4.4.2 au 4.4.5.

#### 4.4.2 ACTIVITES DE GESTION DES RISQUES

Conformément à 4.2 de la présente norme, le FABRICANT doit:

a) évaluer tout retour d'informations, y compris les informations de postproduction, sur le LOGICIEL HERITE concernant les incidents et/ou quasi-incidents tant à l'intérieur de sa propre organisation et/ou que de la part des utilisateurs;

b) effectuer des ACTIVITES DE GESTION DES RISQUES associées à l'utilisation continue du LOGICIEL HERITE, en prenant en considération les aspects suivants:

- intégration du LOGICIEL HERITE dans l'architecture globale du DISPOSITIF MEDICAL;
- validité continue des mesures de MAITRISE DU RISQUE, mises en œuvre dans le cadre du LOGICIEL HERITE;
- identification des SITUATIONS DANGEREUSES associées à l'utilisation continue du LOGICIEL HERITE;
- identification des causes potentielles du LOGICIEL HERITE contribuant à une SITUATION DANGEREUSE;
- définition des mesures de MAITRISE DU RISQUE pour chaque cause potentielle du LOGICIEL HERITE contribuant à une SITUATION DANGEREUSE.

#### 4.4.3 Analyse des lacunes

Sur la base de la classe de sécurité du logiciel du LOGICIEL HERITE (voir 4.3), le FABRICANT doit effectuer une analyse des lacunes des LIVRABLES disponibles par rapport à ceux exigés selon 5.2, 5.3, 5.7 et l'Article 7.

a) Le FABRICANT doit évaluer la validité continue des LIVRABLES disponibles.

b) Lorsque des lacunes sont identifiées, le FABRICANT doit EVALUER la réduction potentielle des RISQUES résultant de la production des LIVRABLES manquants et des ACTIVITES associées.

c) À partir de cette évaluation, le FABRICANT doit déterminer les LIVRABLES à créer et les ACTIVITES associées à effectuer. Au minimum, le LIVRABLE doit comprendre des enregistrements d'essai du SYSTEME LOGICIEL (voir 5.7.5).

NOTE Il convient qu'une telle analyse des lacunes garantisse que les mesures de MAITRISE DU RISQUE, mises en œuvre dans le LOGICIEL HERITE, sont comprises dans les exigences du logiciel.

#### 4.4.4 Activités de comblement des lacunes

a) Le FABRICANT doit établir et exécuter un plan pour produire les LIVRABLES identifiés. Lorsqu'elles sont disponibles, les preuves tangibles peuvent être utilisées pour produire les LIVRABLES exigés sans exécuter les ACTIVITES exigées par 5.2, 5.3, 5.7 et l'Article 7.

NOTE Un plan visant à combler les lacunes identifiées peut être inclus dans un plan de maintenance de logiciel (voir 6.1)

b) Ce plan doit tenir compte de l'utilisation du PROCESSUS de résolution des problèmes pour traiter ceux qui ont été détectés dans le LOGICIEL HERITE et les LIVRABLES conformément à l'Article 9.

c) Les modifications apportées au LOGICIEL HERITE doivent être effectuées conformément à l'Article 6.

#### 4.4.5 Justification de l'utilisation d'un LOGICIEL HERITE

Le FABRICANT doit documenter la VERSION du LOGICIEL HERITE ainsi qu'une justification de la poursuite de l'utilisation du LOGICIEL HERITE sur la base des résultats de 4.4.

NOTE Le respect des dispositions de 4.4 permet une utilisation ultérieure du LOGICIEL HERITE conformément à l'IEC 62304.

## 5 PROCESSUS du développement du logiciel

### 5.1 * Planification du développement du logiciel

#### 5.1.1 Plan de développement du logiciel

*Remplacer, au point e),* «PRODUITS LOGICIELS» *par* «LOGICIELS DE DISPOSITIFS MEDICAUX».

#### 5.1.3 Référence du plan de développement du logiciel à la conception et au développement du SYSTEME

*Remplacer le texte existant du point b) par ce qui suit:*

b) Le FABRICANT doit intégrer ou référencer dans le plan de développement du logiciel les procédures de coordination du développement du logiciel avec le développement du système nécessaires pour satisfaire aux dispositions de 4.1 (telles que l'intégration, la vérification et la validation du système).

#### 5.1.5 Planification de l'intégration du logiciel et des essais d'intégration

*Ajouter la nouvelle note suivante et renuméroter la première note en tant que Note 1:*

NOTE 2 Voir 5.6.

#### 5.1.8 Planification de la documentation

*Supprimer le point c).*

*Ajouter la nouvelle note suivante:*

NOTE Voir l'Article 8 pour une étude sur la gestion de la configuration de la documentation.

#### 5.1.9 Planification de la gestion de configuration du logiciel

*Remplacer, au point c),* «gestion de la configuration du logiciel et les ACTIVITES correspondantes» *par* «ACTIVITES de gestion de configuration du logiciel».

*Ajouter la nouvelle note suivante:*

NOTE Voir l'Article 8.

#### 5.1.10 Éléments annexes à contrôler

*Ajouter la nouvelle note suivante renuméroter la première note en tant que Note 1:*

NOTE 2 Voir l'Article 8.

### 5.1.11 Contrôle de l'ELEMENT DE CONFIGURATION du logiciel avant VERIFICATION

*Remplacer* «sous contrôle documenté de la configuration» *par* «sous gestion de la configuration».

*Ajouter le nouveau paragraphe suivant:*

### 5.1.12 Identification et évitement des défauts logiciels communs

Le FABRICANT doit inclure ou référencer dans le plan de développement du logiciel une procédure pour:

a) identifier des catégories de défauts qui peuvent être présentées à partir de la technologie de programmation sélectionnée et concernant leur SYSTEME LOGICIEL; et

b) fournir des preuves documentées qui montrent que ces défauts ne contribuent pas à un RISQUE inacceptable.

NOTE Pour des exemples de catégories de défauts ou des causes contribuant à des SITUATIONS DANGEREUSES, voir l'Annexe B de l'IEC TR 80002-1:2009.

[Classe B, C]

## 5.2 * Analyse des exigences du logiciel

### 5.2.2 Teneur des exigences du logiciel

*Ajouter à la liste d'exemples de la Note 3 le nouveau point suivant;*

– sûreté du système/protection contre les logiciels malveillants.

*Remplacer le texte existant du point f) par ce qui suit:*

f) exigences de l'interface utilisateur mises en œuvre par le logiciel;

*Remplacer, dans la Note 5* «l'IEC 60601-1-6.» *par* «l'IEC 62366 [21] entre autres (par exemple, l'IEC 60601-1-6 [3]).»

*Remplacer le texte existant du point j) par le nouveau texte et note ci-dessous:*

j) exigences relatives aux aspects réseau TI;

NOTE 9 Exemples comprenant ceux relatifs:

– aux alarmes réseau, avertissements et messages opérateur;

– aux protocoles de réseau;

– à la gestion de l'indisponibilité de services réseau.

*Ajouter la nouvelle note ci-dessous après le point l):*

NOTE 10 Les exigences de a) à l) peuvent se chevaucher.

*Remplacer, dans la Note 8 existante* «l'ISO/IEC 9126-1 [8]» par «Entre autres, l'ISO/IEC 25010 [12]»

### 5.2.3 Intégration des mesures de MAITRISE DU RISQUE dans les exigences du logiciel

*Supprimer la phrase* «pour tenir compte des défaillances matérielles et des éventuels défauts du logiciel*».*

### 5.2.5 Mise à jour des exigences du SYSTEME

*Remplacer le titre existant du paragraphe par ce qui suit:*

### 5.2.5 Mise à jour des exigences

### 5.2.6 Vérification des exigences du logiciel

*Supprimer, dans le texte existant du point d) les mots* «de manière à s'assurer que ces critères sont remplis».

## 5.3 Conception ARCHITECTURALE du logiciel

### 5.3.5 Identification des séparations nécessaires à la MAITRISE DU RISQUE

*Remplacer le texte existant du paragraphe par ce qui suit:*

Le FABRICANT doit identifier toute séparation entre les ELEMENTS LOGICIELS qui est nécessaire pour la MAITRISE DU RISQUE et indiquer la méthode permettant de s'assurer que cette séparation est efficace. [Classe C]

NOTE Un exemple de séparation est l'exécution des ELEMENTS LOGICIELS sur différents processeurs. L'efficacité de la séparation peut être assurée en évitant tout partage de ressources entre les processeurs. D'autres moyens de séparation peuvent être appliqués lorsque l'efficacité peut être assurée par la conception de l'ARCHITECTURE du logiciel (voir B.4.3).

### 5.3.6 VERIFICATION de L'ARCHITECTURE du logiciel

*Ajouter la nouvelle note suivante:*

NOTE Une analyse de TRAÇABILITE de l'ARCHITECTURE par rapport aux exigences du logiciel peut être utilisée afin de satisfaire à l'exigence a).

## 5.4 * Conception détaillée du logiciel

### 5.4.1 Décomposition de L'ARCHITECTURE DES LOGICIELS en UNITES LOGICIELLES

*Remplacer le titre et le texte existants de ce paragraphe par ce qui suit:*

### 5.4.1 Subdivision du logiciel en UNITES LOGICIELLES

Le FABRICANT doit subdiviser le logiciel jusqu'à ce qu'il soit représenté par les UNITES LOGICIELLES. [Classe B, C]

NOTE Certains SYSTEMES LOGICIELS ne sont pas divisés davantage.

### 5.4.2 Élaboration de la conception détaillée de chaque UNITE LOGICIELLE

*Remplacer le texte existant par ce qui suit:*

Le FABRICANT doit documenter une conception avec suffisamment de précisions pour permettre une mise en œuvre correcte de chaque UNITE LOGICIELLE. [Classe C]

### 5.4.3 Élaboration de la conception détaillée pour les interfaces

*Remplacer le texte existant par ce qui suit:*

Le FABRICANT doit documenter une conception des interfaces éventuelles entre l'UNITE LOGICIELLE et les composants externes (matériels ou logiciels), ainsi que pour les interfaces entre les UNITES LOGICIELLES, d'une manière suffisamment détaillée pour mettre en œuvre chaque UNITE LOGICIELLE et ses interfaces correctement. [Classe C]

### 5.4.4 VERIFICATION de la conception détaillée

*Ajouter la nouvelle note suivante:*

NOTE Il est acceptable d'utiliser une analyse de TRAÇABILITE de l'ARCHITECTURE par rapport à la conception détaillée du logiciel afin de satisfaire à l'exigence a).

### 5.5 Mise en œuvre et vérification des UNITES LOGICIELLES

*Remplacer le titre existant de ce paragraphe par ce qui suit:*

### 5.5 Mise en œuvre des UNITES LOGICIELLES

### 5.5.2 Établissement du PROCESSUS DE VERIFICATION DES UNITES LOGICIELLES

*Remplacer le texte existant par ce qui suit:*

Le FABRICANT doit établir des stratégies, méthodes et procédures pour la vérification des UNITES LOGICIELLES. Lorsque la VERIFICATION est effectuée sur la base d'essais, l'adéquation des procédures d'essai doit être EVALUEE. [Classe B, C]

### 5.5.3 Critères d'acceptation de l'UNITE LOGICIELLE

*Remplacer le texte existant du deuxième tiret de la note par ce qui suit:*

– le code du logiciel est-il exempt de toute contradiction avec la conception de l'interface de l'UNITE LOGICIELLE?

### 5.6 Intégration et essai d'intégration du logiciel

### 5.6.2 Vérification de l'intégration du logiciel

*Remplacer le texte existant par ce qui suit:*

Le FABRICANT doit vérifier que les UNITES LOGICIELLES ont été intégrées dans les ELEMENTS LOGICIELS et/ou le SYSTEME LOGICIEL conformément au plan d'intégration (voir 5.1.5) et conserver les preuves de cette vérification. [Classe B, C]

NOTE Cette VERIFICATION vise uniquement à s'assurer que l'intégration a été effectuée conformément au plan. Cette VERIFICATION est en général réalisée sous forme d'inspection.

### 5.6.3 Essai du logiciel intégré

*Remplacer le titre existant par ce qui suit:*

### 5.6.3 Essais d'intégration du logiciel

### 5.6.4 Teneur des essais d'intégration

*Remplacer le titre existant par ce qui suit:*

### 5.6.4 Teneur des essais d'intégration du logiciel

### 5.6.5 Vérification des procédures d'essais d'intégration

*Remplacer le titre et le texte existants par*

### 5.6.5 ÉVALUATION des procédures d'essais d'intégration du logiciel

Le FABRICANT doit EVALUER les procédures d'essais d'intégration pour s'assurer qu'elles sont adéquates. [Classe B, C]

### 5.7 Essais du SYSTEME LOGICIEL

#### 5.7.1 Établissement d'essais pour les exigences du logiciel

*Désigner le texte existant du paragraphe comme point a).*

*Remplacer* «[Classe B, C]» *par* «[Classe A, B, C]».

*Ajouter le nouveau point suivant:*

b) Le FABRICANT doit EVALUER l'adéquation des stratégies de VERIFICATION et des procédures d'essai.

#### 5.7.2 Utilisation du PROCESSUS de résolution des problèmes de logiciel

*Remplacer* «[Classe B, C]» *par* «[Classe A, B, C]».

#### 5.7.3 Contre-essais après modifications

*Remplacer* «[Classe B, C]» *par* «[Classe A, B, C]».

#### 5.7.4 Vérification des essais du SYSTEME LOGICIEL

*Remplacer le titre et texte existants de ce paragraphe par:*

#### 5.7.4 ÉVALUATION des essais du SYSTEME LOGICIEL

Le FABRICANT doit EVALUER si les stratégies de VERIFICATION et les procédures d'essai sont appropriées.

Le FABRICANT doit vérifier que:

a) toutes les exigences du logiciel ont été soumises à des essais ou VERIFIEES par ailleurs;

b) la TRAÇABILITE entre les exigences et les essais du logiciel ou d'autres VERIFICATIONS est consignée; et

c) les résultats d'essai satisfont aux critères de réussite/échec exigés.

[Classe A, B, C]

#### 5.7.5 Teneur des enregistrements d'essai du SYSTEME LOGICIEL

*Remplacer le texte existant par:*

Pour la prise en charge de la répétabilité des essais, le FABRICANT doit documenter:

a) une référence aux procédures du jeu d'essais montrant les actions exigées et les résultats attendus;

b) les résultats d'essai (réussite/échec ainsi que la liste des ANOMALIES);

c) la version du logiciel soumis à l'essai;

d) les configurations d'essai pertinentes pour le matériel et le logiciel;

e) les outils d'essai pertinents;

f) la date de l'essai; et

g) l'identité de la personne responsable de l'exécution de l'essai et de l'enregistrement des résultats de l'essai.

[Classe A, B, C]

### 5.8 Diffusion du logiciel

*Remplacer le titre existant par ce qui suit:*

### 5.8 * DIFFUSION du logiciel pour l'utilisation au niveau du SYSTEME

#### 5.8.1 Assurance de l'achèvement de la VERIFICATION du logiciel

*Remplacer le texte existant par ce qui suit:*

Le FABRICANT doit s'assurer, avant diffusion du logiciel, que toutes les ACTIVITES de VERIFICATION du logiciel sont terminées et que les résultats ont été EVALUES. [Classe A, B, C]

#### 5.8.2 Consignation des ANOMALIES résiduelles connues

*Remplacer* «[Classe B, C]» *par* «[Classe A, B, C]».

#### 5.8.4 Consignation des VERSIONS diffusées

*Remplacer, dans le texte existant, "*PRODUIT LOGICIEL*" par «*LOGICIEL DE DISPOSITIF MEDICAL*".*

#### 5.8.6 Assurance de l'achèvement complet des ACTIVITES et des TACHES

*Remplacer le texte existant par ce qui suit:*

Le FABRICANT doit s'assurer que toutes les ACTIVITES et TACHES du plan de développement (ou du plan de maintenance) du logiciel sont achevées et que la documentation associée est complète. [Classe B, C]

NOTE Voir 5.1.3.b).

#### 5.8.7 Archivage du logiciel

*Remplacer, au point a) «*PRODUIT LOGICIEL*» par «*LOGICIEL DE DISPOSITIF MEDICAL*» et, dans la deuxième partie de la phrase, «dispositif» par «*LOGICIEL DE DISPOSITIF MEDICAL*».*

*Remplacer* «[Classe B, C]» *par* «[Classe A, B, C]».

#### 5.8.8 Assurance de la reproductibilité du logiciel diffusé

*Remplacer le titre existant de ce paragraphe par* ce qui suit:

#### 5.8.8 Assurance de la fiabilité de la distribution du logiciel diffusé

*Remplacer deux fois dans le texte existant le terme «*PRODUIT LOGICIEL*» par «*LOGICIEL DE DISPOSITIF MEDICAL*».*

*Remplacer* «[Classe B, C]» *par* «[Classe A, B, C]».

## 6 Processus de maintenance du logiciel

### 6.1 * Établissement du plan de maintenance du logiciel

*Remplacer, au point e) existant,* «SYSTEME» *par* «SYSTEME LOGICIEL»:

### 6.2 * Analyse des problèmes et des modifications

#### 6.2.1 Consignation et EVALUATION des retours d'information

##### 6.2.1.1 Contrôle des retours d'information

*Remplacer le texte existant par ce qui suit:*

Le FABRICANT doit contrôler les retours d'information sur le LOGICIEL DE DISPOSITIF MEDICAL diffusé pour l'utilisation prévue. [Classe A, B, C]

##### 6.2.1.2 Document et EVALUATION des retours d'information

*Remplacer, dans la première phrase,* «PRODUIT LOGICIEL» *par* «LOGICIEL DE DISPOSITIF MEDICAL».

##### 6.2.1.3 ÉVALUATION des influences des RAPPORTS DE PROBLEME sur la SECURITE

*Remplacer le texte existant par ce qui suit:*

Chaque RAPPORT DE PROBLEME doit être EVALUE afin de déterminer la manière dont il affecte la SECURITE du LOGICIEL DE DISPOSITIF MEDICAL diffusé pour l'utilisation prévue (voir 9.2) et si une modification apportée à ce logiciel est nécessaire pour remédier au problème. [Classe A, B, C]

#### 6.2.2 Utilisation du PROCESSUS de résolution des problèmes du logiciel

*Remplacer le texte existant de la note par ce qui suit:*

NOTE Un problème peut révéler qu'un SYSTEME LOGICIEL ou un ELEMENT LOGICIEL n'a pas été placé dans la classe de sécurité du logiciel correcte. Le processus de résolution des problèmes peut impliquer des modifications de la classe de sécurité du logiciel. Une fois le PROCESSUS terminé, il convient de connaître et de documenter toute modification de classe de sécurité dans le SYSTEME LOGICIEL ou ses ELEMENTS LOGICIELS.

#### 6.2.3 Analyse des DEMANDES DE MODIFICATION

*Remplacer le texte existant par ce qui suit:*

En plus de l'analyse exigée par l'Article 9, le FABRICANT doit analyser chaque DEMANDE DE MODIFICATION afin d'évaluer ses effets sur l'organisation, sur les LOGICIELS DE DISPOSITIFS MEDICAUX diffusés pour l'utilisation prévue, ainsi que les SYSTEMES auxquels il est interfacé. [Classe A, B, C]

#### 6.2.4 Approbation des DEMANDES DE MODIFICATION

*Remplacer* «PRODUITS LOGICIELS» *par* «LOGICIELS DE DISPOSITIFS MEDICAUX».

#### 6.2.5 Communication aux utilisateurs et aux organismes de réglementation

*Remplacer* «PRODUITS LOGICIELS» *par* «LOGICIELS DE DISPOSITIFS MEDICAUX» (3 fois).

### 6.3 * Mise en œuvre de la modification

#### 6.3.1 Utilisation d'un PROCESSUS établi pour mettre en œuvre la modification

*Remplacer le texte existant par ce qui suit:*

Le FABRICANT doit identifier et exécuter toute ACTIVITE de l'Article 5 nécessitant d'être répétée du fait de la modification. [Classe A, B, C]

NOTE Pour les exigences concernant la GESTION DES RISQUES des modifications du logiciel, voir 7.4.

### 6.3.2 Rediffusion du SYSTEME LOGICIEL modifié

*Remplacer le texte existant par:*

Le FABRICANT doit diffuser les modifications conformément à 5.8. [Classe A, B, C]

NOTE Les modifications peuvent être diffusées dans le cadre d'une rediffusion complète d'un SYSTEME LOGICIEL ou comme un kit de modifications comprenant les ELEMENTS LOGICIELS modifiés et les outils nécessaires pour installer les modifications pour un SYSTEME LOGICIEL existant.

## 7 PROCESSUS de GESTION des RISQUES du logiciel

### 7.1.5 Consignation des séquences d'événements

*Supprimer ce paragraphe.*

### 7.2 Mesures de MAITRISE DU RISQUE

### 7.2.1 Définition des mesures de MAITRISE DU RISQUE

*Remplacer le texte existant par ce qui suit:*

Pour chaque cas documenté dans le DOSSIER DE GESTION DES RISQUES lorsqu'un ELEMENT LOGICIEL peut contribuer à une SITUATION DANGEREUSE, le FABRICANT doit définir et documenter les mesures de MAITRISE DU RISQUE conformément à l'ISO 14971.

[Classe B, C]

NOTE Les mesures de MAITRISE DU RISQUE peuvent être mises en œuvre dans le matériel, dans le logiciel, dans l'environnement de travail ou comme une instruction destinée à l'utilisateur.

### 7.2.2 Mesures de MAITRISE DU RISQUE mises en œuvre dans le logiciel

*Remplacer le texte existant du point b) par ce qui suit:*

b) attribuer à chaque élément logiciel contribuant à la mise en œuvre d'une mesure de maîtrise du risque une classe de sécurité du logiciel sur la base du RISQUE que contrôle la mesure de MAITRISE DU RISQUE (voir 4.3 a)); et

### 7.3 Vérification des mesures de maîtrise du risque

### 7.3.1 Vérification des mesures de MAITRISE DU RISQUE

*Ajouter le texte suivant après la première phrase et avant la mention* «[Classe B, C]» *existantes:*

Le FABRICANT doit revoir la mesure de MAITRISE DU RISQUE et déterminer si ce RISQUE peut entraîner une nouvelle SITUATION DANGEREUSE.

### 7.3.2 Consignation de toutes nouvelles séquences d'événements

*Supprimer le texte et le titre existants et ajouter* “Non utilisé”.

# 8 * PROCESSUS de gestion de configuration du logiciel

## 8.1 * Identification de la configuration

### 8.1.1 Établissement des moyens d'identification des ELEMENTS DE CONFIGURATION

*Remplacer le texte existant par ce qui suit:*

Le FABRICANT doit établir un plan pour l'identification univoque des ELEMENTS DE CONFIGURATION et leurs VERSIONS à maîtriser conformément à la planification du développement et de la configuration spécifiée en 5.1. [Classe A, B, C]

### 8.1.2 Identification des logiciels SOUP

*Supprimer, à la suite du point c), les mots* «de chaque ELEMENT DE CONFIGURATION de logiciel SOUP à utiliser.»

## 8.2 * Maîtrise des modifications

### 8.2.1 Approbation des DEMANDES DE MODIFICATION

*Ajouter, après le terme «*ELEMENTS DE CONFIGURATION» *les mots suivants: «*identifiés comme éléments à contrôler conformément à 8.1».

*Remplacer, dans la Note 2,* «voir 5.1.1 e)» *par* «voir 5.1.1 d)».

### 8.2.4 Prévision des moyens de TRAÇABILITE de la modification

*Remplacer le texte existant par ce qui suit:*

Le FABRICANT doit tenir à jour des enregistrements des relations et des dépendances entre:

a) la DEMANDE DE MODIFICATION;

b) le RAPPORT DE PROBLEME pertinent; et

c) l'approbation de la DEMANDE DE MODIFICATION.

[Classe A, B, C]

# 9 PROCESSUS de résolution de problème logiciel

## 9.1 Élaboration des RAPPORTS DE PROBLEME

*Remplacer le texte existant par ce qui suit:*

Le FABRICANT doit rédiger un RAPPORT DE PROBLEME pour chaque problème détecté dans un LOGICIEL DE DISPOSITIF MEDICAL. Les RAPPORTS DE PROBLEME doivent inclure un énoncé de criticité (par exemple, les effets sur les performances, la SECURITE ou la SURETE), ainsi que toute autre information qui peut faciliter la résolution du problème (par exemple, les dispositifs concernés et les accessoires pris en charge concernés).

[Classe A, B, C]

NOTE Les problèmes peuvent être décelés avant ou après diffusion, à l'intérieur ou à l'extérieur de l'organisation du FABRICANT.

### 9.5 Conservation des enregistrements

*Supprimer, à la fin du deuxième alinéa,* «(voir 7.4)».

### 9.7 VERIFICATION de la résolution des problèmes du logiciel

*Remplacer, au point c),* «PRODUITS LOGICIELS» *par* «LOGICIELS DE DISPOSITIFS MEDICAUX».

## Annexe A – Justification des exigences de la présente norme

### A.1 Justification du raisonnement

*Remplacer, dans la première phrase du quatrième alinéa, le terme «situation dangereuse» par le même terme en petites majuscules:* «SITUATION DANGEREUSE».

*Remplacer, dans la deuxième phrase du quatrième alinéa, le terme* «DANGERS» *par* «SITUATIONS DANGEREUSES».

*Supprimer, dans la deuxième phrase du dernier alinéa, les mots:* «qui ne peuvent par définition donner lieu à un PHENOMENE DANGEREUX».

*Supprimer, dans la troisième phrase du dernier alinéa, le mot* «facilement».

### Tableau A.1 – Récapitulatif des exigences par classe de sécurité du logiciel

*Remplacer le tableau existant par ce qui suit:*

**Tableau A.1 – Récapitulatif des exigences par classe de sécurité du logiciel**

| Articles et paragraphes | | Classe A | Classe B | Clases C |
|---|---|---|---|---|
| Article 4 | Toutes les exigences | X | X | X |
| 5.1 | 5.1.1, 5.1.2, 5.1.3, 5.1.6, 5.1.7, 5.1.8, 5.1.9, | X | X | X |
| | 5.1.5, 5.1.10, 5.1.11, 5.1.12 | | X | X |
| | 5.1.4 | | | X |
| 5.2 | 5.2.1, 5.2.2, 5.2.4, 5.2.5, 5.2.6 | X | X | X |
| | 5.2.3 | | X | X |
| 5.3 | 5.3.1, 5.3.2, 5.3.3, 5.3.4, 5.3.6 | | X | X |
| | 5.3.5 | | | X |
| 5.4 | 5.4.1 | | X | X |
| | 5.4.2, 5.4.3, 5.4.4 | | | X |
| 5.5 | 5.5.1 | X | X | X |
| | 5.5.2, 5.5.3, 5.5.5 | | X | X |
| | 5.5.4 | | | X |
| 5.6 | Toutes les exigences | | X | X |
| 5.7 | Toutes les exigences | X | X | X |
| 5.8 | 5.8.1, 5.8.2, 5.8.4, 5.8.7, 5.8.8 | X | X | X |
| | 5.8.3, 5.8.5, 5.8.6 | | X | X |
| Article 6 | Toutes les exigences | X | X | X |
| 7.1 | Toutes les exigences | | X | X |
| 7.2 | Toutes les exigences | | X | X |
| 7.3 | Toutes les exigences | | X | X |
| 7.4 | 7.4.1 | X | X | X |
| | 7.4.2, 7.4.3 | | X | X |
| Article 8 | Toutes les exigences | X | X | X |
| Article 9 | Toutes les exigences | X | X | X |

## Annexe B – Lignes directrices relatives aux dispositions de la présente norme

### B.1.1 Objet

*Remplacer, dans la deuxième phrase du premier alinéa, le terme* «PRODUITS LOGICIELS» *par* «LOGICIELS DE DISPOSITIFS MEDICAUX».

### B.1.2 Domaine d'application

*Remplacer, dans la dernière phrase du deuxième alinéa,* «PROCESSUS DE GESTION DES RISQUES du DISPOSITIF MEDICAL» *par* «du PROCESSUS global de GESTION DES RISQUES du DISPOSITIF MEDICAL».

*Remplacer, dans la deuxième phrase du troisième alinéa, le terme* «PRODUITS LOGICIELS» *par* «LOGICIELS DE DISPOSITIFS MEDICAUX».

### B.3 Termes et définitions

*Ajouter, à la fin de la deuxième phrase du deuxième alinéa, après* «à part entière» *le groupe de mots* «, qui devient alors un LOGICIEL DE DISPOSITIF MEDICAL».

### B.4 Exigences générales

#### B.4.2 GESTION DES RISQUES

*Remplacer, dans la dernière phrase du deuxième alinéa, le terme* «DANGERS» PAR «SITUATIONS DANGEREUSES».

#### B.4.3 Classification de sécurité du logiciel

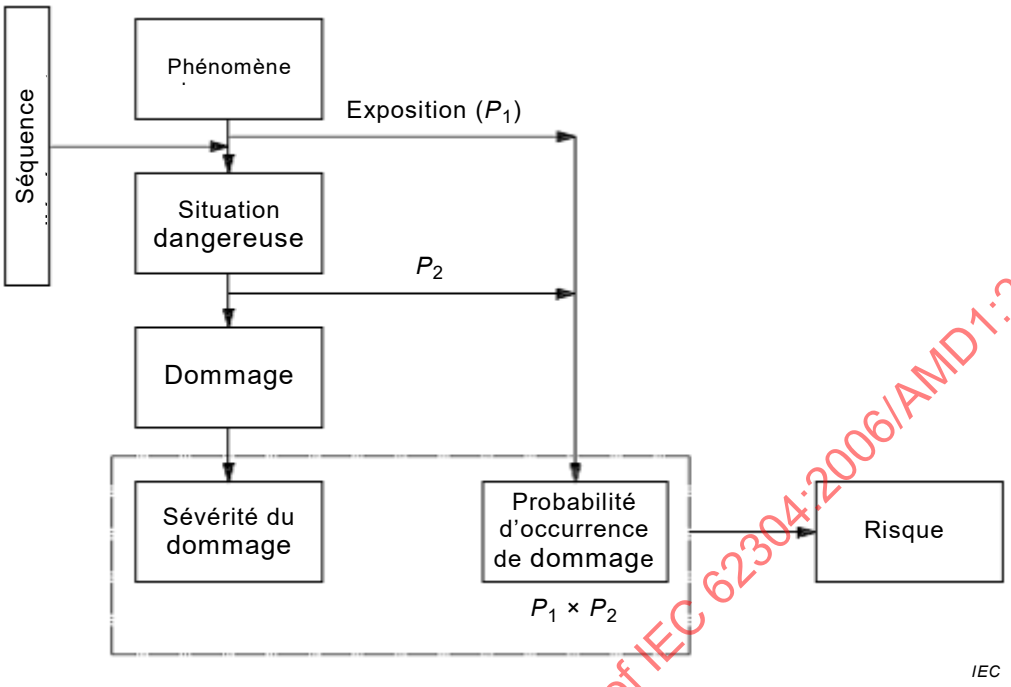
*Remplacer le deuxième alinéa existant par ce qui suit:*

Le RISQUE est défini comme étant une combinaison de la gravité du DOMMAGE et de la probabilité de sa survenance. Cependant, il n'existe pas de consensus quant à une méthode d'estimation quantitative de la probabilité d'une défaillance logicielle. Lorsque le logiciel est présent dans une séquence ou une combinaison d'événements conduisant à une SITUATION DANGEREUSE, la probabilité d'une défaillance logicielle ne peut pas être prise en considération dans l'estimation du RISQUE de la SITUATION DANGEREUSE. Dans ce cas, il est judicieux d'envisager le cas le plus défavorable, et il convient de définir la probabilité d'une défaillance logicielle sur 1. Lorsqu'il est possible d'estimer la probabilité des autres événements de la séquence (ce qui peut être le cas s'ils ne sont pas logiciels), cette probabilité peut être utilisée pour déterminer celle de la SITUATION DANGEREUSE ($P_1$ à la Figure B. 2).

Dans de nombreux cas, toutefois, il peut ne pas être possible d'estimer la probabilité des autres événements de la séquence, et il convient d'EVALUER le RISQUE en fonction de la nature du DOMMAGE seule (il convient de définir la probabilité de la SITUATION DANGEREUSE sur 1). Dans ce cas, il convient que l'ESTIMATION DU RISQUE soit focalisée sur la GRAVITE du DOMMAGE qui résulte de la SITUATION DANGEREUSE. Des classements subjectifs de probabilité peuvent également être attribués selon les connaissances cliniques, afin de distinguer les défaillances qu'un clinicien discernera probablement de celles qui risquent de ne pas être détectées et sont les plus susceptibles de causer des DOMMAGES.

Les estimations de la probabilité d'une SITUATION DANGEREUSE entraînant des DOMMAGES ($P_2$ à la Figure B.2) exigent généralement des connaissances cliniques pour pouvoir distinguer les

SITUATIONS DANGEREUSES où les pratiques cliniques sont susceptibles de prévenir les DOMMAGES des SITUATIONS DANGEREUSES qui sont plus susceptibles de causer des DOMMAGES.




NOTE $P_1$ est la probabilité qu'une situation dangereuse se produise

$P_2$ est la probabilité qu'une situation dangereuse entraîne un dommage

**Figure B.2 – Illustration de la relation entre PHENOMENE DANGEREUX (DANGER), séquence d'événements, SITUATION DANGEREUSE et DOMMAGE – tirée de l'Annexe E de l'ISO 14971:2007**

*Ajouter, à la fin du douzième alinéa existant* («Il convient que l'ARCHITECTURE du logiciel favorise…»), *les deux nouvelles phrases suivantes:*

La différenciation n'est pas limitée à la séparation physique (processeur ou partitions de mémoire), mais inclut tout mécanisme empêchant un ELEMENT LOGICIEL de nuire à un autre. L'adéquation d'une différenciation est déterminée par les RISQUES impliqués et la justification exigée à documenter.

*Remplacer, dans la première phrase du dernier alinéa,* «logiciel de DISPOSITIF MEDICAL» *par* «LOGICIEL DE DISPOSITIF MEDICAL» (le terme entier en petites majuscules). *(La correction de la première phrase du dernier alinéa ne concerne que le texte anglais)*

*Remplacer, dans la troisième phrase du dernier alinéa, le terme* «DANGERS» *par* «SITUATIONS DANGEREUSES», *deux fois.*

*Remplacer la dernière phrase du dernier alinéa par ce qui suit:*

Si la différenciation n'est pas possible entre les ELEMENTS LOGICIELS X et Y, l'ELEMENT LOGICIEL X doit alors être classé dans la classe C de sécurité du logiciel.

*Ajouter le nouveau paragraphe suivant:*

### B.4.4 LOGICIEL HERITE

Le paragraphe 4.4 établit un processus pour l'application de la présente norme aux LOGICIELS HERITES. Certains pays peuvent exiger du FABRICANT qu'il démontre la conformité à la norme pour obtenir l'approbation réglementaire du LOGICIEL DE DISPOSITIF MEDICAL, même si ce logiciel a été conçu avant la version actuelle de la norme (LOGICIEL HERITE). Dans ce cas, les exigences de 4.4 fournissent au FABRICANT une méthode lui permettant de démontrer que le LOGICIEL HERITE est conforme à la norme.

Un FABRICANT peut déterminer que la documentation rétrospective d'un cycle de vie de développement déjà terminé qui s'est déroulé isolément ne se traduit pas par la réduction du RISQUE associé à l'utilisation du produit. Le processus entraîne l'identification d'un sous-ensemble d'ACTIVITES défini dans la présente norme, qui n'entraîne pas la réduction du RISQUE. Autres buts implicites du processus:

– il convient que les ACTIVITES exigées et la documentation résultante se fondent et utilisent, dans la mesure du possible, la documentation existante, et
– il convient qu'un FABRICANT exploite les ressources le plus efficacement possible en vue de réduire les RISQUES.

Outre un plan qui identifie le sous-ensemble d'ACTIVITES à exécuter, le processus produit également des preuves tangibles en faveur de la poursuite de l'utilisation du LOGICIEL HERITE en toute sécurité, ainsi qu'une justification résumée de la présente conclusion.

Les RISQUES associés à la poursuite prévue de l'utilisation du LOGICIEL HERITE dépendent du contexte dans lequel le LOGICIEL HERITE est utilisé pour créer un SYSTEME LOGICIEL. Le FABRICANT documente tous les DANGERS liés au DISPOSITIF MEDICAL identifiés et associés au LOGICIEL HERITE.

Le paragraphe 4.4 exige une évaluation complète des données de postproduction sur le terrain disponibles obtenues pour le LOGICIEL HERITE au cours de la période pendant laquelle il a été en production et en utilisation. Les sources types de données de postproduction incluent:

– les événements indésirables imputables au dispositif,
– les retours d'information des utilisateurs du dispositif, et
– les ANOMALIES découvertes par le FABRICANT.

Bien qu'il n'existe pas de consensus quant à une méthode prospective d'estimation quantitative de la probabilité d'une défaillance logicielle, de telles informations peuvent être disponibles pour les LOGICIELS HERITES selon leur utilisation et l'EVALUATION des données de postproduction. Dans ce cas, s'il est possible d'estimer quantitativement la probabilité d'événements dans la séquence, une valeur quantitative peut être utilisée pour exprimer la probabilité d'occurrence de toute la séquence d'événements. Si une telle estimation quantitative n'est pas possible, il est judicieux d'envisager la probabilité du cas le plus défavorable, et il convient de définir la probabilité d'une défaillance logicielle sur 1.

La détermination par le FABRICANT de la manière dont le LOGICIEL HERITE sera utilisé dans l'ARCHITECTURE globale du SYSTEME du DISPOSITIF MEDICAL contribue à l'appréciation du RISQUE. Les RISQUES à prendre en considération varient en conséquence.

– Lorsqu'un LOGICIEL HERITE a été utilisé de manière sûre et fiable, et que le FABRICANT souhaite continuer à l'utiliser, la justification de ce choix repose principalement sur l'appréciation du RISQUE à partir des enregistrements de postproduction.
– Lorsqu'un LOGICIEL HERITE est réutilisé en vue de créer un nouveau SYSTEME LOGICIEL, l'utilisation prévue du LOGICIEL HERITE peut être différente de celle qui était prévue à l'origine. Dans ce cas, l'appréciation du RISQUE doit tenir compte de l'ensemble modifié des SITUATIONS DANGEREUSES pouvant résulter de défaillances du LOGICIEL HERITE.

– Un LOGICIEL HERITE peut être réutilisé dans un but similaire à l'utilisation prévue, mais intégré dans un nouveau SYSTEME LOGICIEL. Dans ce cas, il convient que l'appréciation du RISQUE tienne compte de la modification des mesures de MAITRISE DU RISQUE architectural conformément à 5.3.

Si un LOGICIEL HERITE est modifié et utilisé dans un nouveau SYSTEME LOGICIEL, il convient que le FABRICANT tienne compte du fait que les enregistrements existants sur le fonctionnement sûr et fiable peuvent être invalidés suite aux modifications.

Il convient que les modifications apportées aux LOGICIELS HERITES soient effectuées conformément aux Articles 4 à 9 de la présente norme, y compris l'évaluation des répercussions sur les mesures de MAITRISE DU RISQUE selon 7.4. Dans le cas d'un LOGICIEL HERITE, les mesures existantes de MAITRISE DU RISQUE peuvent ne pas être parfaitement documentées. Il convient alors d'EVALUER soigneusement les effets potentiels des modifications à partir des enregistrements documentés de la conception disponibles, ainsi que de l'expertise des individus connaissant le système.

Selon 4.4, le FABRICANT procède à une analyse des lacunes afin de déterminer la documentation disponible, y compris les preuves tangibles des TACHES accomplies pendant le développement du LOGICIEL HERITE et en les comparant aux dispositions de 5.2, 5.3, 5.7 et de l'Article 7. Les étapes types de cette analyse des lacunes comprennent

a) l'identification du LOGICIEL HERITE, y compris sa VERSION, son numéro de révision et tout autre moyen exigé pour une identification claire;

b) l'EVALUATION des LIVRABLES existants correspondant à ceux exigés par 5.2, 5.3, 5.7 et l'Article 7;

c) l'EVALUATION des preuves tangibles disponibles, en documentant le modèle du cycle de vie de développement du logiciel précédemment appliqué (selon le cas);

d) l'EVALUATION de l'adéquation de la documentation existante sur la GESTION DES RISQUES, en tenant compte de l'ISO 14971.

En tenant compte de l'analyse des lacunes effectuée, le FABRICANT EVALUE la réduction potentielle des RISQUES résultant de la production des LIVRABLES manquants et des ACTIVITES associées, et établit un plan en vue d'exécuter les ACTIVITES et de produire des LIVRABLES pour combler ces lacunes.

Il convient que la réduction des RISQUES équilibre l'avantage de l'application du processus de développement du logiciel tel que défini à l'Article 5, d'une part, et la possibilité qu'une modification du LOGICIEL HERITE sans une parfaite connaissance de l'historique de son développement introduise de nouveaux défauts susceptibles d'accroître le risque, d'autre part. L'évaluation après coup de certains éléments de l'Article 5 peut révéler qu'ils ne réduisent que peu les RISQUES, voire pas du tout. Par exemple, une vérification de la conception détaillée et des unités réduit principalement les RISQUES pendant le processus de développement d'un nouveau logiciel ou de refactorisation d'un logiciel existant. Si ces objectifs ne sont pas planifiés, exécuter ces ACTIVITES isolément peut produire de la documentation, mais sans conduire à une réduction des RISQUES.

Au minimum, le plan de comblement des lacunes permet de pallier le manque d'enregistrements des essais du SYSTEME LOGICIEL. Si ces documents n'existent pas ou n'apportent aucune justification de la poursuite de l'utilisation du LOGICIEL HERITE, il convient que le plan de comblement des lacunes comprenne l'établissement d'exigences vis-à-vis du SYSTEME LOGICIEL au niveau fonctionnel, conformément à 5.2, mais aussi au niveau des essais, conformément à 5.7.

La justification documentée de la poursuite de l'utilisation du LOGICIEL HERITE repose sur les preuves tangibles disponibles ainsi que sur l'analyse effectuée au cours de l'appréciation du RISQUE et de l'établissement d'un plan de comblement des lacunes adapté au contexte de la réutilisation du LOGICIEL HERITE.

Cette justification souligne la sûreté et la fiabilité du LOGICIEL HERITE dans le contexte de la réutilisation prévue, en tenant compte à la fois des enregistrements de postproduction disponibles pour le LOGICIEL HERITE et des MESURES DE MAITRISE DU RISQUE effectuées en comblant les lacunes des processus.

Lorsque le LOGICIEL HERITE a été réutilisé conformément à 4.4, les parties pour lesquelles il subsiste des lacunes en termes de LIVRABLES restent un élément du LOGICIEL HERITE et peuvent être prises en considération pour une nouvelle réutilisation conformément à 4.4. Lorsque les lacunes en termes de livrables sont comblées par modification du LOGICIEL HERITE, il convient d'effectuer les modifications conformément aux Articles 4 à 9 de la présente norme.

## B.5 PROCESSUS du développement du logiciel

### B.5.1 Planification du développement du logiciel

*Remplacer, dans les troisième et quatrième phrases du deuxième alinéa, le terme «*PRODUIT LOGICIEL DE DISPOSITIF MEDICAL» *par «*LOGICIEL DE DISPOSITIF MEDICAL».

### B.5.3 Conception ARCHITECTURALE du logiciel

*Remplacer la première phrase du premier alinéa par ce qui suit:*

Cette ACTIVITE exige que le FABRICANT définisse les principaux composants structurels du logiciel et identifie les principales responsabilités, leurs caractéristiques visibles de l'extérieur ainsi que les relations qui existent entre eux.

*Ajouter, à la fin de la troisième phrase du premier alinéa* «(voir 5.3.5 et B.4.3)».

*Remplacer la première phrase du deuxième alinéa par ce qui suit:*

La classification de sécurité du logiciel des ELEMENTS LOGICIELS pendant l'ACTIVITE d'ARCHITECTURE du logiciel génère une base pour le choix ultérieur des PROCESSUS logiciels.

### B.5.4 Conception détaillée du logiciel

*Remplacer, dans la troisième phrase du premier alinéa, les mots* «Nous avons» *par* «La présente norme a».

*Remplacer les huitième et neuvième phrases du premier alinéa par ce qui suit:*

Il est nécessaire de définir la conception des UNITES LOGICIELLES et des interfaces de manière suffisamment détaillée pour pouvoir effectuer une VERIFICATION objective de sa SECURITE et de son efficacité lorsque ceci peut être assuré à l'aide d'autres exigences ou de la documentation de conception.

*Ajouter, à la fin du premier alinéa, la nouvelle phrase suivante;*

La conception détaillée doit également traiter de l'architecture du LOGICIEL DE DISPOSITIF MEDICAL.

*Ajouter, après la quatrième phrase du troisième alinéa, la nouvelle phrase suivante:*

La VERIFICATION de la conception donne l'assurance qu'elle met en œuvre l'ARCHITECTURE logicielle et qu'elle est exempte de toute contradiction avec l'ARCHITECTURE logicielle.

### B.5.6 Intégration et essai d'intégration du logiciel

*Remplacer, dans la dernière phrase du sixième alinéa, «*un PRODUIT LOGICIEL*» par «*LOGICIEL DE DISPOSITIF MEDICAL».

### B.5.7 Essais du SYSTEME LOGICIEL

*Ajouter, à la fin du cinquième alinéa* «(Voir B.6.3).».

*Remplacer, dans la première phrase du septième alinéa (avant-dernier alinéa), le terme «*DANGERS*» par «*RISQUES».

## B.6 PROCESSUS de maintenance du logiciel

### B.6.2 Analyse des problèmes et des modifications

*Remplacer, dans la cinquième phrase du premier alinéa, le terme "*DANGER*" par "*SITUATION DANGEREUSE*" à deux reprises, pour que la phrase soit libellée comme suit:*

Il est également important de vérifier que le logiciel modifié n'entraîne pas une SITUATION DANGEREUSE ou n'occulte pas un RISQUE alors que précédemment il n'entraînait pas une SITUATION DANGEREUSE ou n'occultait pas de RISQUES dans le logiciel.

*Remplacer, dans la première phrase du troisième alinéa,* «PRODUIT LOGICIEL» *par* «LOGICIEL DE DISPOSITIF MEDICAL».

Remplacer, au deuxième tiret du troisième alinéa, les mots «les PRODUITS LOGICIELS sont revalidés» par «le LOGICIEL DE DISPOSITIF MEDICAL est revalidé».

Remplacer, au troisième tiret du troisième alinéa, «PRODUITS LOGICIELS» par «LOGICIEL DE DISPOSITIF MEDICAL».

### B.6.3 Mise en œuvre de la modification

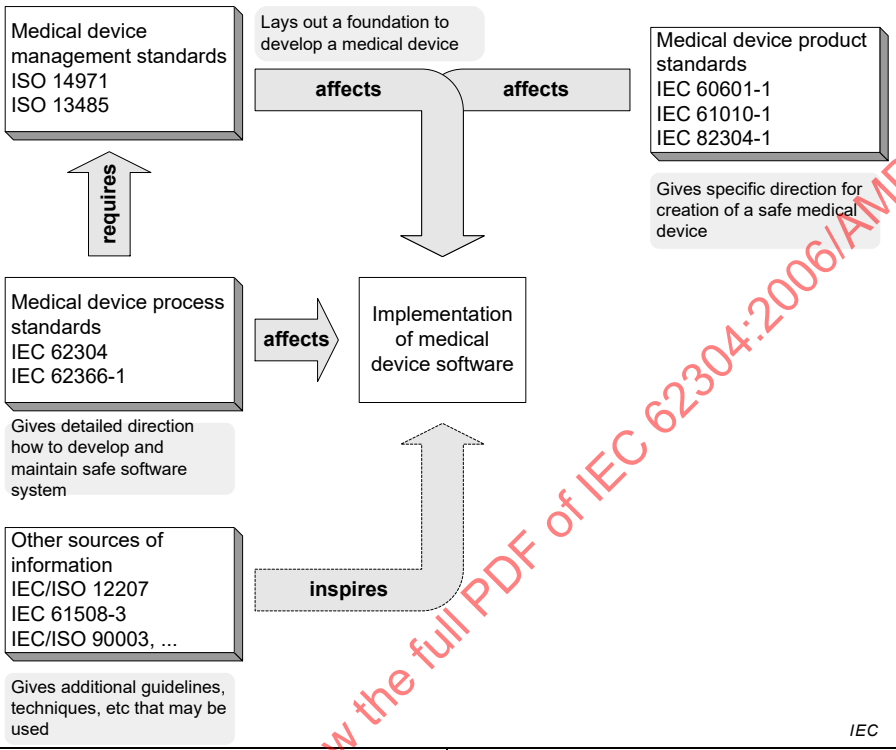
*Insérer, après la quatrième phrase du texte existant, les trois nouvelles phrases suivantes:*

L'analyse et des essais de régression sont effectués pour garantir qu'une modification n'a pas créé de problèmes ailleurs dans le LOGICIEL DE DISPOSITIF MEDICAL. L'analyse de régression consiste à déterminer les effets d'une modification à partir de la revue de la documentation concernée (par exemple, spécification des exigences du logiciel, spécification de la conception du logiciel, code source, plans d'essai, jeux d'essai, scripts d'essai, etc.) afin d'identifier les essais de régression nécessaires à effectuer. Un essai de régression consiste à refaire les jeux d'essais précédemment exécutés correctement par un programme et à comparer le résultat actuel au précédent afin de détecter les éventuels effets indésirables d'une modification logicielle.

## Annexe C – Relations avec d'autres normes

### C.1 Généralités

*Remplacer la Figure C.1 existante par la nouvelle figure suivante, dans laquelle les références à l'IEC 62366-1 et à l'IEC 82304-1 ont été ajoutées aux normes de processus de dispositifs médicaux et aux normes de produits de dispositifs médicaux, respectivement:*




| Anglais | Français |
|---|---|
| Medical device management standards<br>ISO 14971<br>ISO 13485 | Normes de gestion de dispositifs médicaux<br>ISO 14971<br>ISO 13485 |
| Lays out a foundation to develop a medical device | Établissent les bases pour développer un dispositif médical |
| affects | conditionnent |
| Medical device product standards<br>IEC 60601-1<br>IEC 61010-1<br>IEC 82304-1 | Normes de produits de dispositifs médicaux<br>IEC 60601-1<br>IEC 61010-1<br>IEC 82304-1 |
| Gives specific direction for creation of a safe medical device | Donnent des instructions spécifiques pour la création d'un dispositif médical sûr |
| Medical device process standards<br>IEC 62304<br>IEC 62366-1 | Normes de processus de dispositifs médicaux<br>IEC 62304<br>IEC 62366-1 |
| Implementation of medical device software | Mise en œuvre du logiciel de dispositif médical |
| Gives detailed direction how to develop and maintain safe software system | Donnent des instructions détaillées sur la manière de développer et d'assurer la maintenance d'un système logiciel sûr |
| Other sources of information<br>IEC/ISO 12207<br>IEC 61508-3<br>IEC/ISO 90003, … | Autres sources d'information<br>IEC/ISO 12207<br>IEC 61508-3<br>IEC/ISO 90003, … |
| inspires | influent |
| Gives additional guidelines, techniques, etc. that may be used | Donnent des lignes directrices supplémentaires, des techniques, etc. qui peuvent être utilisées |

**Figure C.1 – Relation des principales normes de DISPOSITIFS MEDICAUX avec l'IEC 62304**