# IEC 62304

Edition 1.0 2015-06

# INTERNATIONAL STANDARD

AMENDMENT 1

## Medical device software – Software life cycle processes

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 It also gives information on projects, replaced and withdrawn publications.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 more than 30 000 terms and definitions in English and French, with equivalent terms in 15 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

**IEC Glossary - std.iec.ch/glossary**
More than 60 000 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.

# IEC 62304

Edition 1.0 2015-06

# INTERNATIONAL STANDARD

AMENDMENT 1

## Medical device software – Software life cycle processes

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 11.040

ISBN 978-2-8322-2720-6

# 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.

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 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.

A bilingual version of this publication may be issued at a later date.

# INTRODUCTION TO THE AMENDMENT

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).

[1] In preparation.

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.

## 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]

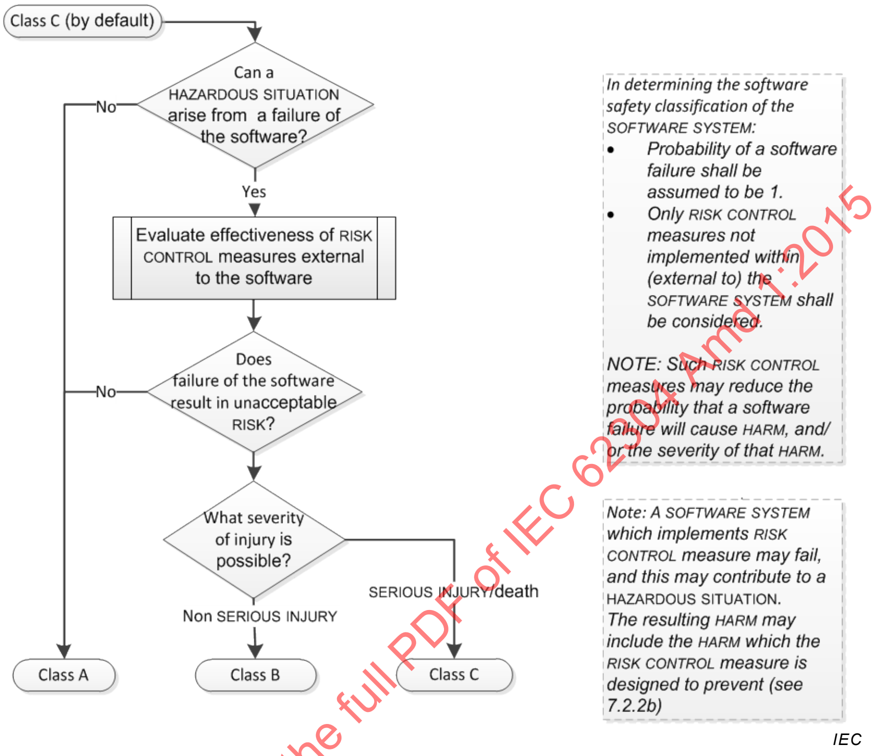
### 3.40
**RISK EVALUATION**
PROCESS of comparing the estimated RISK against given RISK criteria to determine the acceptability of the RISK

[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.


Class C (by default)
Can a HAZARDOUS SITUATION arise from a failure of the software?
No
Yes
Evaluate effectiveness of RISK CONTROL measures external to the software
Does failure of the software result in unacceptable RISK?
No
What severity of injury is possible?
SERIOUS INJURY/death
Non SERIOUS INJURY
Class A
Class B
Class C
IEC

*In determining the software safety classification of the SOFTWARE SYSTEM:*

- *Probability of a software failure shall be assumed to be 1.*
- *Only RISK CONTROL measures not implemented within (external to) the SOFTWARE SYSTEM shall be considered.*

*NOTE: Such RISK CONTROL measures may reduce the probability that a software failure will cause HARM, and/ or the severity of that HARM.*

*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)*

**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*".*

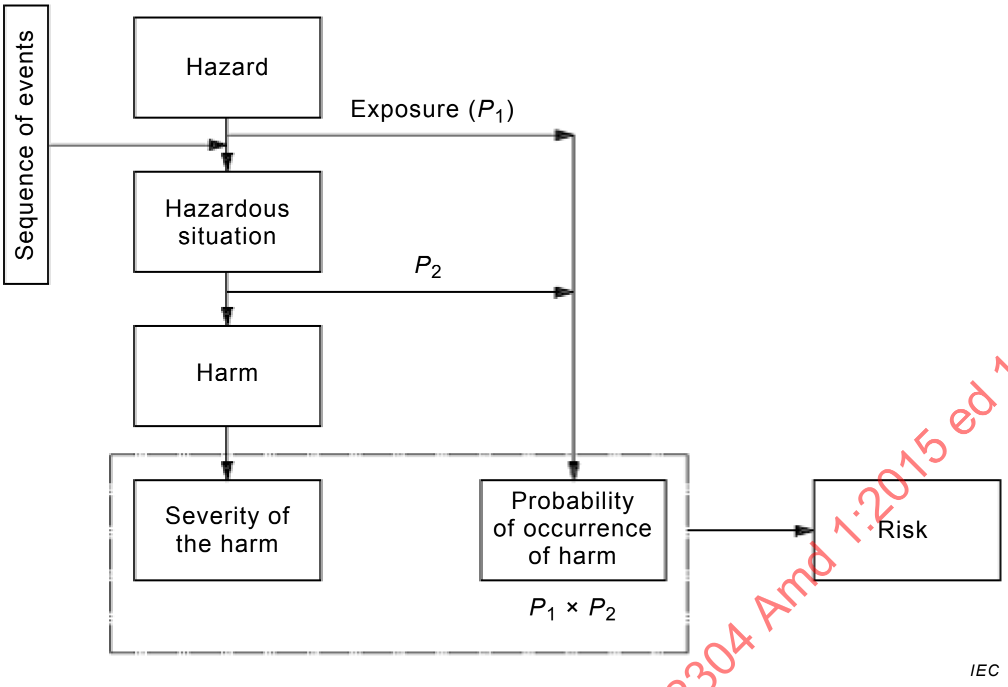
### B.4.3 Software safety classification

*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:*

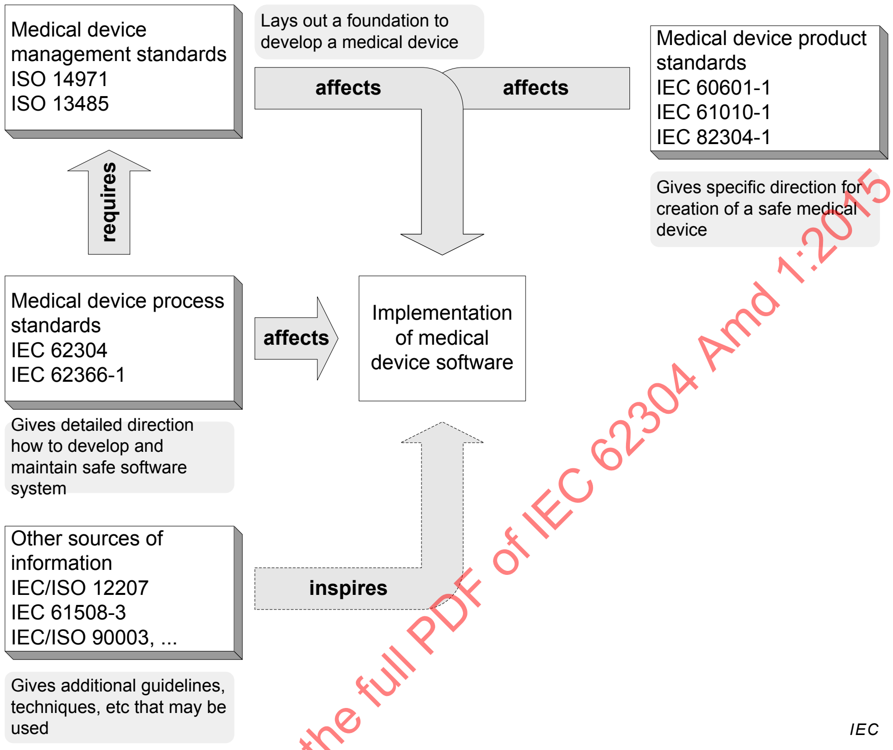

Medical device management standards
ISO 14971
ISO 13485
Lays out a foundation to develop a medical device
affects
affects
Medical device product standards
IEC 60601-1
IEC 61010-1
IEC 82304-1
Gives specific direction for creation of a safe medical device
requires
Medical device process standards
IEC 62304
IEC 62366-1
affects
Implementation of medical device software
Gives detailed direction how to develop and maintain safe software system
Other sources of information
IEC/ISO 12207
IEC 61508-3
IEC/ISO 90003, ...
inspires
Gives additional guidelines, techniques, etc that may be used
IEC

**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 RISK CONTROL measures implemented in software<br>7.3.1 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:*