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 It also gives information on projects, replaced and withdrawn publications.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

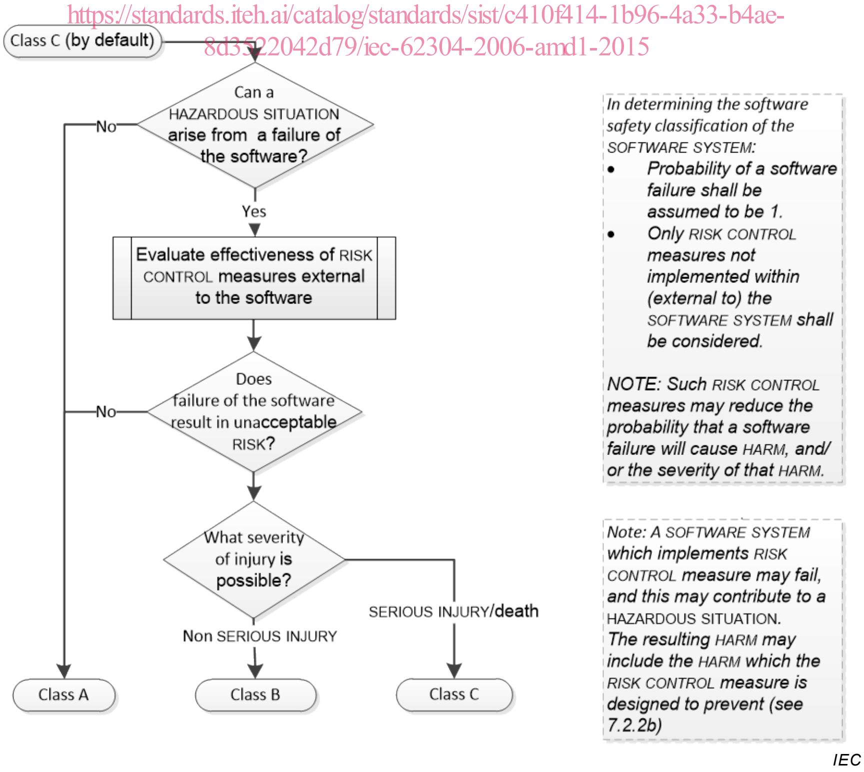
PROCESS of comparing the estimated RISK against given RISK criteria to determine the acceptability of the RISK

[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.


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?
Non SERIOUS INJURY
SERIOUS INJURY/death
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: