NORME INTERNATIONALE

INTERNATIONAL STANDARD

CEI
IEC
62304

Première édition
First edition
2006-05

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

Medical device software – Software life cycle processes

Numéro de référence
Reference number
CEI/IEC 62304:2006

Numérotation des publications

Depuis le 1er janvier 1997, les publications de la CEI sont numérotées à partir de 60000. Ainsi, la CEI 34-1 devient la CEI 60034-1.

Editions consolidées

Les versions consolidées de certaines publications de la CEI incorporant les amendements sont disponibles. Par exemple, les numéros d'édition 1.0, 1.1 et 1.2 indiquent respectivement la publication de base, la publication de base incorporant l'amendement 1, et la publication de base incorporant les amendements 1 et 2.

Informations supplémentaires sur les publications de la CEI

Le contenu technique des publications de la CEI est constamment revu par la CEI afin qu'il reflète l'état actuel de la technique. Des renseignements relatifs à cette publication, y compris sa validité, sont disponibles dans le Catalogue des publications de la CEI (voir ci-dessous) en plus des nouvelles éditions, amendements et corrigenda. Des informations sur les sujets à l'étude et l'avancement des travaux entrepris par le comité d'études qui a élaboré cette publication, ainsi que la liste des publications parues, sont également disponibles par l'intermédiaire de:

- **Site web de la CEI (www.iec.ch)**
- **Catalogue des publications de la CEI**

  Le catalogue en ligne sur le site web de la CEI (www.iec.ch/searchpub) vous permet de faire des recherches en utilisant de nombreux critères, comprenant des recherches textuelles, par comité d'études ou date de publication. Des informations en ligne sont également disponibles sur les nouvelles publications, les publications remplacées ou retirées, ainsi que sur les corrigenda.
- **IEC Just Published**

  Ce résumé des dernières publications parues (www.iec.ch/online_news/justpub) est aussi disponible par courrier électronique. Veuillez prendre contact avec le Service client (voir ci-dessous) pour plus d'informations.
- **Service clients**

  Si vous avez des questions au sujet de cette publication ou avez besoin de renseignements supplémentaires, prenez contact avec le Service clients:

  Email: custserv@iec.ch
  Tél: +41 22 919 02 11
  Fax: +41 22 919 03 00

Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site (www.iec.ch)**
- **Catalogue of IEC publications**

  The on-line catalogue on the IEC web site (www.iec.ch/searchpub) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.
- **IEC Just Published**

  This summary of recently issued publications (www.iec.ch/online_news/justpub) is also available by email. Please contact the Customer Service Centre (see below) for further information.
- **Customer Service Centre**

  If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

  Email: custserv@iec.ch
  Tel: +41 22 919 02 11
  Fax: +41 22 919 03 00

.

NORME INTERNATIONALE

INTERNATIONAL STANDARD

CEI
IEC
62304

Première édition
First edition
2006-05

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

Medical device software – Software life cycle processes

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch

CODE PRIX
PRICE CODE XC

Pour prix, voir catalogue en vigueur
For price, see current catalogue

SOMMAIRE

CONTENTS

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

LOGICIELS DE DISPOSITIFS MÉDICAUX – PROCESSUS DU CYCLE DE VIE DU LOGICIEL

AVANT-PROPOS

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

La Norme Internationale CEI 62304 a été établie par un groupe de travail mixte du sous-comité 62A: Aspects généraux des équipements utilisés en pratique médicale, du comité technique 62 de la CEI: Equipements électriques dans la pratique médicale et du comité technique 210 de l'ISO, management de la qualité et aspects généraux correspondants des dispositifs médicaux. Le Tableau C.5 a été préparé par le Comité Technique mixte ISO/CEI 1/SC7, Ingénierie du logiciel et du système.

Elle est publiée comme norme portant un double logo.

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
62A/523/FDIS	62A/528/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme. A l'ISO, la norme a été approuvée par 23 membres participants sur les 23 ayant voté.

INTERNATIONAL ELECTROTECHNICAL COMMISSION

MEDICAL DEVICE SOFTWARE – SOFTWARE LIFE CYCLE PROCESSES

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62304 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. Table C.5 was prepared by ISO/IEC JTC 1/SC 7, Software and system engineering.

It is published as a dual logo standard.

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

FDIS	Report on voting
62A/523/FDIS	62A/528/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table. In ISO, the standard has been approved by 23 P-members out of 23 having cast a vote.

La présente publication a été préparée conformément aux directives de l'ISO/CEI, Partie 2.

Les polices de caractère suivantes sont utilisées dans la présente norme:

- exigences et définitions: en caractères romains;
- des éléments d'information apparaissant hors des tableaux tels que les notes, les exemples et les références: en petits caractères. Le texte normatif des tableaux est également en petits caractères;
- les termes utilisés partout dans la présente norme, qui ont été définis dans l'article 3 et énumérés également dans l'index: en petites majuscules.

Lorsqu'un astérisque (*) est utilisé comme premier caractère d'un titre ou au début d'un paragraphe, il indique que des lignes directrices relatives à cet élément sont fournies en Annexe B.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de maintenance indiquée sur le site web de la CEI sous «http://webstore.iec.ch» dans les données relatives à la publication recherchée. A cette date, la publication sera

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

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

In this standard the following print types are used:

- requirements and definitions: in roman type;
- informative material appearing outside of tables, such as notes, examples and references: in smaller type. Normative text of tables is also in a smaller type;
- terms used throughout this standard that have been defined in Clause 3 and also given in the index: in small capitals.

An asterisk (*) as the first character of a title or at the beginning of a paragraph indicates that there is guidance related to that item in Annex B.

The committee has decided that the contents of this publication will remain unchanged until the maintenance result date indicated on the IEC web site under “http://webstore.iec.ch” in the data related to the specific publication. At this date, the publication will be

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

INTRODUCTION

Le logiciel fait souvent partie intégrante de la technologie des DISPOSITIFS MÉDICAUX. La détermination de la SÉCURITÉ et de l'efficacité d'un DISPOSITIF MÉDICAL comportant un logiciel exige que soit connu ce qu'il est prévu que le logiciel accomplisse et qu'il soit démontré que son utilisation remplit ces objectifs sans entraîner de RISQUES inacceptables.

La présente norme fournit un cadre pour les PROCESSUS du cycle de vie en définissant les ACTIVITES et TÂCHES nécessaires à la conception et à la maintenance en toute SÉCURITÉ des LOGICIELS DE DISPOSITIFS MÉDICAUX. La présente norme fournit les exigences applicables à chaque PROCESSUS du cycle de vie. Chaque PROCESSUS du cycle de vie est en outre divisé en un ensemble D'ACTIVITÉS dont la plupart sont ensuite divisées en un ensemble de TÂCHES.

On suppose par principe que les LOGICIELS DE DISPOSITIFS MÉDICAUX sont développés et maintenus dans le cadre d'un système de management de la qualité (voir 4.1) et d'un système de GESTION DES RISQUES (voir 4.2). Le PROCESSUS DE GESTION DES RISQUES est déjà parfaitement traité dans la Norme Internationale ISO 14971. En conséquence, la norme CEI 62304 tire profit de cet avantage par une simple référence normative à l'ISO 14971. Cependant, pour les logiciels, des exigences supplémentaires mineures de GESTION DES RISQUES sont nécessaires, notamment dans le domaine de l'identification des facteurs contributifs des logiciels en termes de DANGER. Ces exigences sont résumées et introduites dans l'Article 7, PROCESSUS DE GESTION DES RISQUES liés au logiciel.

L'éventuelle contribution d'un logiciel à un DANGER donné est déterminée lors de L'ACTIVITÉ d'identification des DANGERS du PROCESSUS DE GESTION DES RISQUES. LES DANGERS qui pourraient être indirectement induits par les logiciels (par exemple la fourniture d'informations propres à induire en erreur qui pourrait donner lieu à l'administration d'un traitement inadéquat) doivent être pris en compte lorsqu'il s'agit de déterminer si le logiciel est un facteur contributif. La décision d'utiliser le logiciel pour maîtriser les RISQUES est prise lors de L'ACTIVITÉ DE MAÎTRISE DES RISQUES du PROCESSUS DE GESTION DES RISQUES. Le PROCESSUS DE GESTION DES RISQUES lié au logiciel prescrit dans la présente norme doit être intégré au PROCESSUS DE GESTION DES RISQUES lié au dispositif conformément à l'ISO 14971.

Le PROCESSUS de développement des logiciels couvre un certain nombre d'ACTIVITÉS. Ces ACTIVITÉS sont illustrées en Figure 1 et décrites dans l'Article 5. Parce qu'il est notoire que de nombreux incidents sur le terrain sont liés à l'entretien ou à la maintenance des SYSTÈMES DE DISPOSITIFS MÉDICAUX comprenant des mises à jour et des mises à niveau inadéquates du logiciel, on considère que le PROCESSUS de maintenance des logiciels est aussi important que le PROCESSUS de développement des logiciels. Le PROCESSUS de maintenance des logiciels est très similaire au PROCESSUS de développement des logiciels. Cela est illustré en Figure 2 et décrit dans l'Article 6.

INTRODUCTION

Software is often an integral part of MEDICAL DEVICE technology. Establishing the SAFETY and effectiveness of a MEDICAL DEVICE containing software requires knowledge of what the software is intended to do and demonstration that the use of the software fulfils those intentions without causing any unacceptable RISKS.

This standard provides a framework of life cycle PROCESSES with ACTIVITIES and TASKS necessary for the safe design and maintenance of MEDICAL DEVICE SOFTWARE. This standard provides requirements for each life cycle PROCESS. Each life cycle PROCESS is further divided into a set of ACTIVITIES, with most ACTIVITIES further divided into a set of TASKS.

As a basic foundation it is assumed that MEDICAL DEVICE SOFTWARE is developed and maintained within a quality management system (see 4.1) and a RISK MANAGEMENT system (see 4.2). The RISK MANAGEMENT PROCESS is already very well addressed by the International Standard ISO 14971. Therefore IEC 62304 makes use of this advantage simply by a normative reference to ISO 14971. Some minor additional RISK MANAGEMENT requirements are needed for software, especially in the area of identification of contributing software factors related to HAZARDS. These requirements are summarized and captured in Clause 7 as the software RISK MANAGEMENT PROCESS.

Whether software is a contributing factor to a HAZARD is determined during the HAZARD identification ACTIVITY of the RISK MANAGEMENT PROCESS. HAZARDS that could be indirectly caused by software (for example, by providing misleading information that could cause inappropriate treatment to be administered) need to be considered when determining whether software is a contributing factor. The decision to use software to control RISK is made during the RISK CONTROL ACTIVITY of the RISK MANAGEMENT PROCESS. The software RISK MANAGEMENT PROCESS required in this standard has to be embedded in the device RISK MANAGEMENT PROCESS according to ISO 14971.

The software development PROCESS consists of a number of ACTIVITIES. These ACTIVITIES are shown in Figure 1 and described in Clause 5. Because many incidents in the field are related to service or maintenance of MEDICAL DEVICE SYSTEMS including inappropriate software updates and upgrades, the software maintenance PROCESS is considered to be as important as the software development PROCESS. The software maintenance PROCESS is very similar to the software development PROCESS. It is shown in Figure 2 and described in Clause 6.

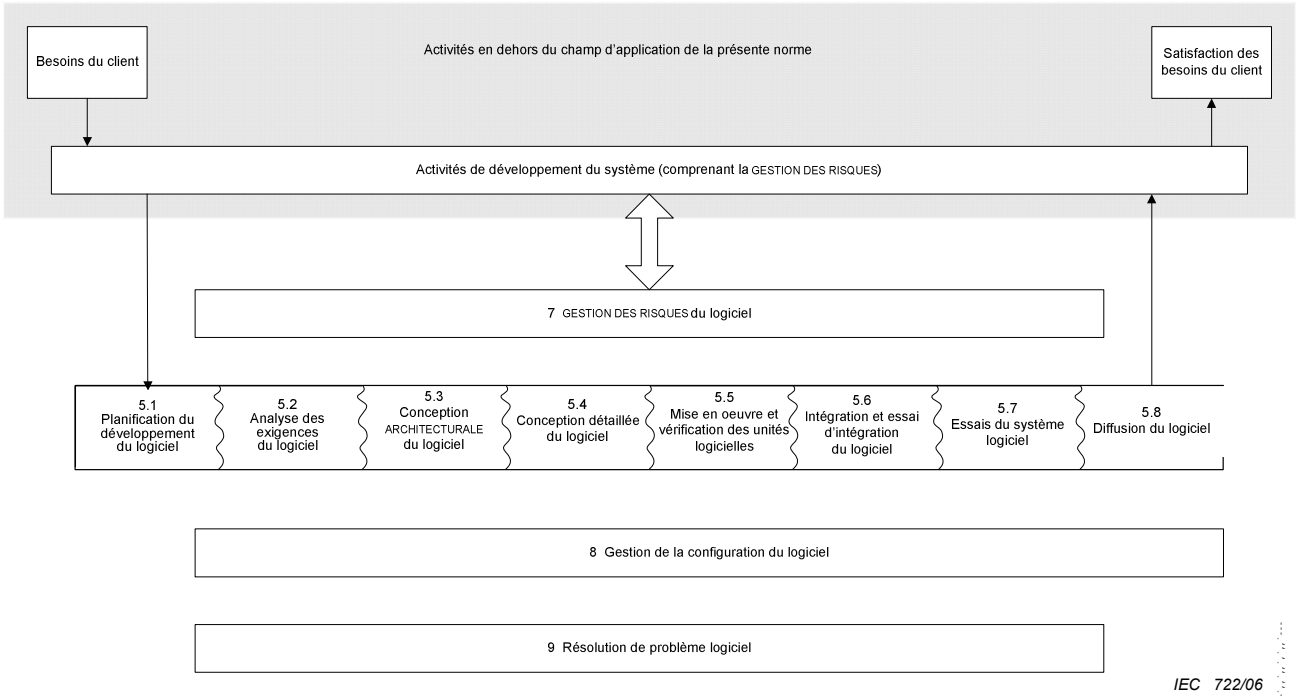

Figure 1 – Présentation générale des PROCESSUS et ACTIVITÉS de développement de logiciels

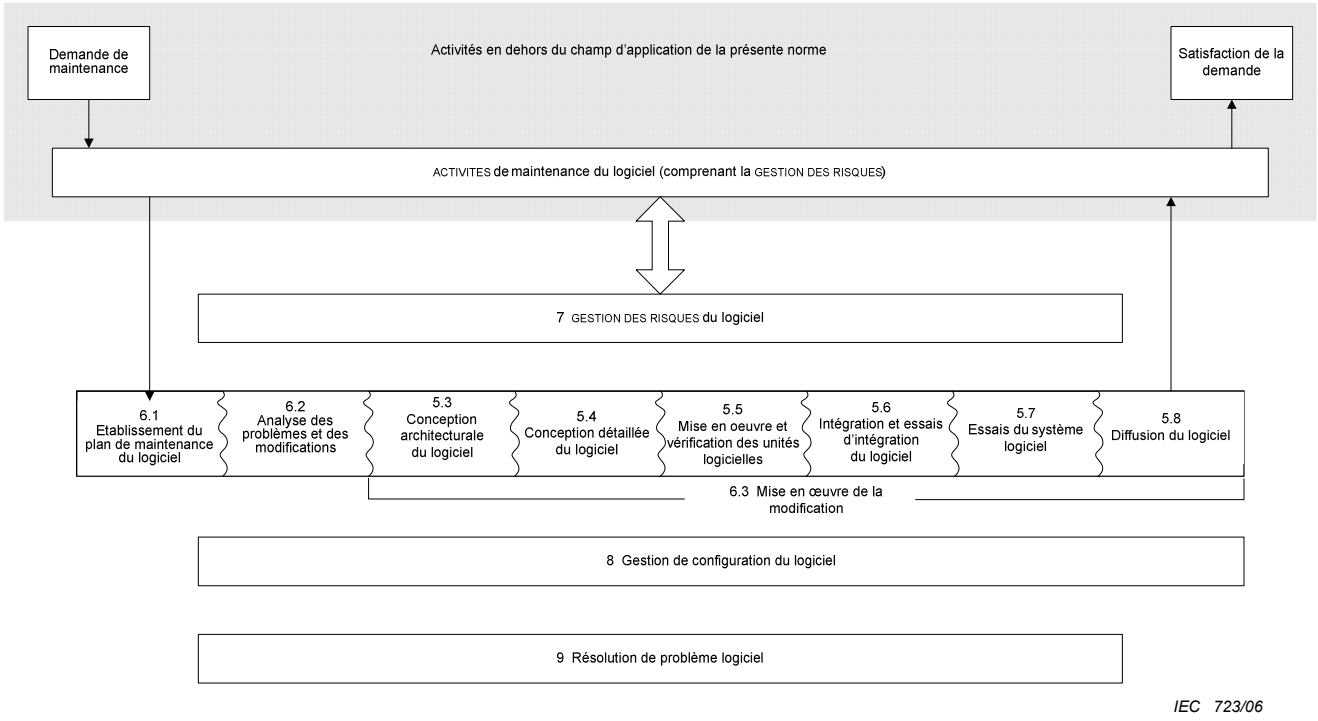

Figure 2 – Présentation générale des PROCESSUS et ACTIVITÉS de maintenance de logiciels

La présente norme identifie deux PROCESSUS additionnels considérés comme essentiels pour le développement de LOGICIELS DE DISPOSITIFS MÉDICAUX sûrs. Il s'agit du PROCESSUS de gestion de la configuration du logiciel (Article 8) et du PROCESSUS de résolution des problèmes de logiciel (Article 9).

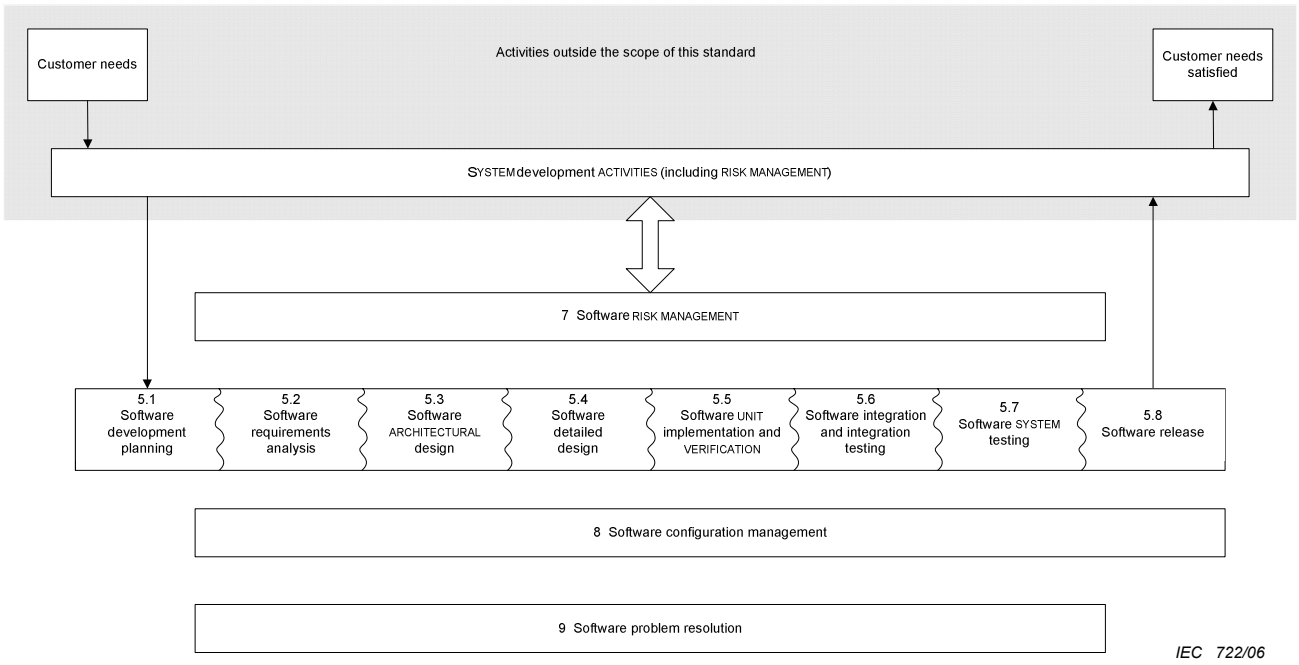

Figure 1 – Overview of software development PROCESSES and ACTIVITIES

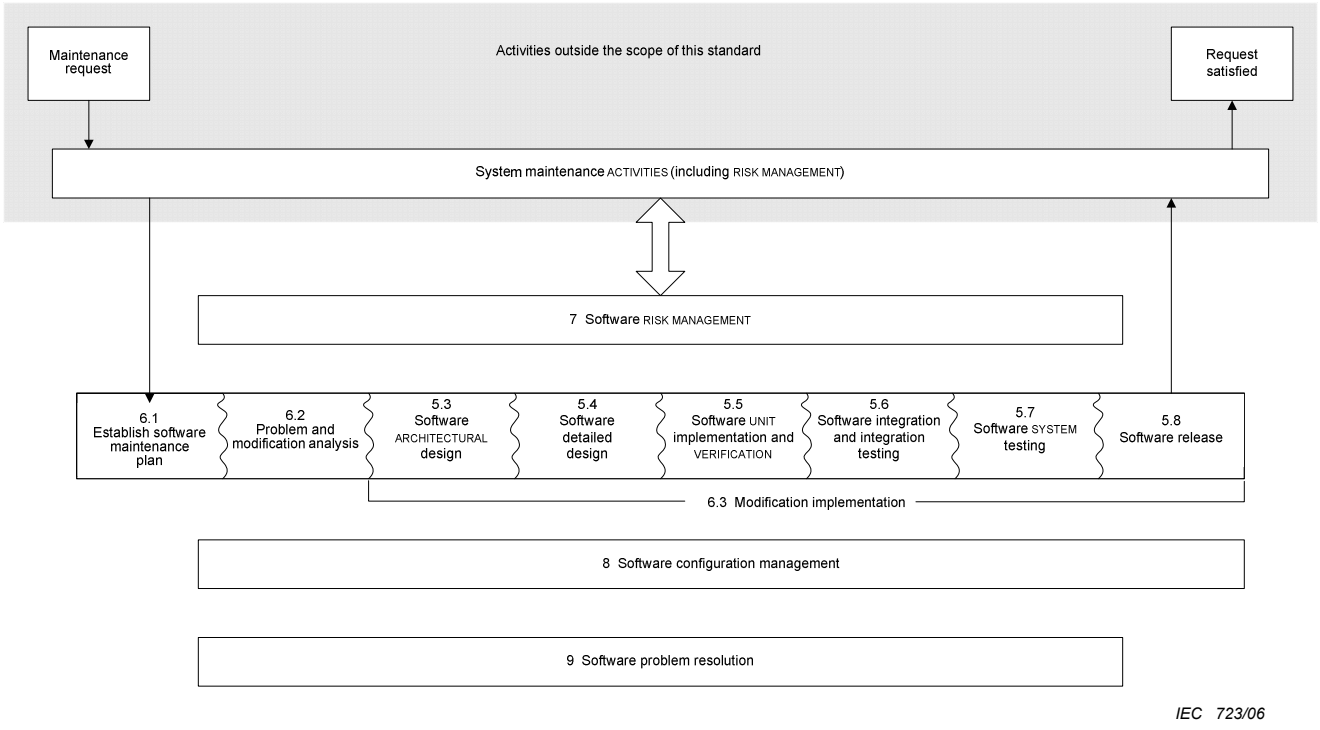

Figure 2 – Overview of software maintenance PROCESSES and ACTIVITIES

This standard identifies two additional PROCESSES considered essential for developing safe MEDICAL DEVICE SOFTWARE. They are the software configuration management PROCESS (Clause 8) and the software problem resolution PROCESS (Clause 9).

La présente norme ne prescrit aucune structure organisationnelle pour le FABRICANT et n'entend pas spécifier quelle organisation doit réaliser tel ou tel PROCESSUS, ACTIVITÉ ou TÂCHE. La présente norme exige uniquement que le PROCESSUS, l'ACTIVITÉ ou la TÂCHE soit mené à bien pour assurer la conformité à la présente norme.

La présente norme ne prescrit pas de désignation, de format ou de contenu explicite de la documentation à produire. Elle exige que les TÂCHES soient documentées, mais c'est à l'utilisateur de décider de la manière dont la documentation correspondante doit être présentée.

La présente norme ne prescrit pas un modèle de cycle de vie spécifique. Il incombe aux utilisateurs de la présente norme de choisir un modèle de cycle de vie pour un projet de logiciel et de faire correspondre les PROCESSUS, ACTIVITÉS et TÂCHES définis dans la présente norme avec ce modèle.

L'Annexe A fournit une justification des articles de la présente norme. L'Annexe B donne des conseils relatifs aux dispositions de la présente norme.

Pour les besoins de la présente norme:

- «doit» signifie qu'une exigence donnée est obligatoire pour être conforme à la présente norme;
- «il convient de – est recommandé» signifie qu'une exigence donnée est recommandée mais n'est pas obligatoire pour être conforme à la présente norme;
- «peut – est admis» est utilisé pour décrire une manière autorisée pour être conforme à une prescription donnée;
- «établir» signifie définir, documenter et mettre en œuvre; et
- Lorsque la présente norme utilise l'expression «si nécessaire» ou «le cas échéant», conjointement à un PROCESSUS, une ACTIVITÉ, une TÂCHE ou un résultat exigés, cela signifie que le FABRICANT doit utiliser le PROCESSUS, l'ACTIVITÉ, la TÂCHE ou le résultat et dans le cas contraire il doit justifier sa décision par écrit.

This standard does not specify an organizational structure for the MANUFACTURER or which part of the organization is to perform which PROCESS, ACTIVITY, or TASK. This standard requires only that the PROCESS, ACTIVITY, or TASK be completed to establish compliance with this standard.

This standard does not prescribe the name, format, or explicit content of the documentation to be produced. This standard requires documentation of TASKS, but the decision of how to package this documentation is left to the user of the standard.

This standard does not prescribe a specific life cycle model. The users of this standard are responsible for selecting a life cycle model for the software project and for mapping the PROCESSES, ACTIVITIES, and TASKS in this standard onto that model.

Annex A provides rationale for the clauses of this standard. Annex B provides guidance on the provisions of this standard.

For the purposes of this standard:

- “shall” means that compliance with a requirement is mandatory for compliance with this standard;
- “should” means that compliance with a requirement is recommended but is not mandatory for compliance with this standard;
- “may” is used to describe a permissible way to achieve compliance with a requirement;
- “establish” means to define, document, and implement; and
- where this standard uses the term “as appropriate” in conjunction with a required PROCESS, ACTIVITY, TASK or output, the intention is that the MANUFACTURER shall use the PROCESS, ACTIVITY, TASK or output unless the MANUFACTURER can document a justification for not so doing.

LOGICIELS DE DISPOSITIFS MÉDICAUX – PROCESSUS DU CYCLE DE VIE DU LOGICIEL

1 Domaine d'application

1.1 *Objet

La présente norme définit les exigences du cycle de vie des LOGICIELS DE DISPOSITIFS MÉDICAUX. L'ensemble des PROCESSUS, ACTIVITÉS et TÂCHES décrit dans la présente norme constitue un cadre commun pour les PROCESSUS du cycle de vie des LOGICIELS DE DISPOSITIFS MÉDICAUX.

1.2 * Domaine d'application

La présente norme s'applique au développement et à la maintenance des LOGICIELS DE DISPOSITIFS MÉDICAUX.

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

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

1.3 Relations avec d'autres normes

La présente norme couvrant le cycle de vie des LOGICIELS DE DISPOSITIFS MÉDICAUX doit être utilisée conjointement à d'autres normes pertinentes pour le développement d'un DISPOSITIF MÉDICAL. L'Annexe C présente les relations existant entre la présente norme et d'autres normes pertinentes.

1.4 Conformité

La conformité à la présente norme est définie comme la mise en œuvre de tous les PROCESSUS, ACTIVITÉS et TÂCHES identifiés dans la présente norme en fonction de la classe de sécurité.

NOTE Les classes de sécurité du logiciel assignées à chaque exigence sont identifiées dans le texte normatif suivant l'exigence.

La conformité est déterminée par inspection de toute documentation exigée par la présente norme y compris le DOSSIER DE GESTION DES RISQUES et l'évaluation des PROCESSUS, ACTIVITÉS et TÂCHES requis pour la classe de SÉCURITÉ du logiciel. Voir l'Annexe D.

NOTE 1 Cette évaluation peut être effectuée par audit interne ou externe.

NOTE 2 Même lorsque les PROCESSUS, ACTIVITÉS et TÂCHES sont effectivement réalisés, il existe une certaine flexibilité dans les méthodes de mise en œuvre de ces PROCESSUS et d'exécution de ces ACTIVITÉS et TÂCHES.

NOTE 3 Lorsqu'une éventuelle exigence comporte la mention «le cas échéant» ou «si nécessaire» et qu'elle n'est pas réalisée, une justification écrite est nécessaire pour cette évaluation.

NOTE 4 Dans la version anglaise de l'ISO/CEI 12207 le terme «conformance» est utilisé pour «conformité», alors que dans la version anglaise de la présente norme, on utilise le terme «compliance».

MEDICAL DEVICE SOFTWARE – SOFTWARE LIFE CYCLE PROCESSES

1 Scope

1.1 * Purpose

This standard defines the life cycle requirements for MEDICAL DEVICE SOFTWARE. The set of PROCESSES, ACTIVITIES, and TASKS described in this standard establishes a common framework for MEDICAL DEVICE SOFTWARE life cycle PROCESSES.

1.2 * Field of application

This standard applies to the development and maintenance of MEDICAL DEVICE SOFTWARE.

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.

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

1.3 Relationship to other standards

This MEDICAL DEVICE SOFTWARE life cycle standard is to be used together with other appropriate standards when developing a MEDICAL DEVICE. Annex C shows the relationship between this standard and other relevant standards.

1.4 Compliance

Compliance with this standard is defined as implementing all of the PROCESSES, ACTIVITIES, and TASKS identified in this standard in accordance with the software safety class.

NOTE The software safety classes assigned to each requirement are identified in the normative text following the requirement.

Compliance is determined by inspection of all documentation required by this standard including the RISK MANAGEMENT FILE, and assessment of the PROCESSES, ACTIVITIES and TASKS required for the software safety class. See Annex D.

NOTE 1 This assessment could be carried out by internal or external audit.

NOTE 2 Although the specified PROCESSES, ACTIVITIES, and TASKS are performed, flexibility exists in the methods of implementing these PROCESSES and performing these ACTIVITIES and TASKS.

NOTE 3 Where any requirements contain “as appropriate” and were not performed, documentation for the justification is necessary for this assessment.

NOTE 4 The term “conformance” is used in ISO/IEC 12207 where the term “compliance” is used in this standard.

2 * Références normatives

Les documents de référence suivants sont indispensables pour l'application du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

ISO 14971, *Dispositifs médicaux – Application de la gestion des risques aux dispositifs médicaux*.

3 *Termes et définitions

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

3.1
ACTIVITÉ
ensemble d'une ou de plusieurs TÂCHES corrélées ou interactives

3.2
ANOMALIE
tout état qui s'écarte de ce qui est attendu sur la base des spécifications des exigences, des documents de conception, des normes, etc., ou qui ne correspond pas à la perception ou à l'expérience d'un individu donné. Les ANOMALIES peuvent être décelées, sans limitation aucune, pendant la revue, l'essai, l'analyse, la compilation ou l'utilisation des PRODUITS LOGICIELS ou de la documentation applicable

[IEEE 1044:1993, définition 3.1]

3.3
ARCHITECTURE
structure organisationnelle d'un SYSTÈME ou d'un composant

[IEEE 610.12:1990]

3.4
DEMANDE DE MODIFICATION
spécification écrite d'une modification à effectuer sur un PRODUIT LOGICIEL

3.5
ÉLÉMENT DE CONFIGURATION
entité qui peut être identifiée de manière univoque en un point de référence donné

NOTE Basé sur l'ISO/CEI 12207:1995, définition 3.6

3.6
LIVRABLE
résultat ou élément de sortie requis (y compris la documentation) d'une ACTIVITÉ ou d'une TÂCHE

3.7
ÉVALUATION
détermination systématique de l'étendue à laquelle l'entité répond aux critères spécifiés

[ISO/CEI 12207:1995, définition 3.9]

2 * Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 14971, *Medical devices – Application of risk management to medical devices*.

3 * Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
ACTIVITY
a set of one or more interrelated or interacting TASKS

3.2
ANOMALY
any condition that deviates from the expected based on requirements specifications, design documents, standards, etc. or from someone's perceptions or experiences. ANOMALIES may be found during, but not limited to, the review, test, analysis, compilation, or use of SOFTWARE PRODUCTS or applicable documentation

[IEEE 1044:1993, definition 3.1]

3.3
ARCHITECTURE
organizational structure of a SYSTEM or component

[IEEE 610.12:1990]

3.4
CHANGE REQUEST
a documented specification of a change to be made to a SOFTWARE PRODUCT

3.5
CONFIGURATION ITEM
entity that can be uniquely identified at a given reference point

NOTE Based on ISO/IEC 12207:1995, definition 3.6.

3.6
DELIVERABLE
required result or output (includes documentation) of an ACTIVITY or TASK

3.7
EVALUATION
a systematic determination of the extent to which an entity meets its specified criteria

[ISO/IEC 12207:1995, definition 3.9]

3.8
DOMMAGE
blessure physique ou atteinte à la santé des personnes ou atteinte aux biens ou à l'environnement

[ISO/CEI Guide 51:1999, définition 3.3]

3.9
PHÉNOMÈNE DANGEREUX (DANGER)
source potentielle de DOMMAGE

[ISO/CEI Guide 51:1999, définition 3.5]

3.10
FABRICANT
personne physique ou morale chargée de la conception, de la fabrication, du conditionnement ou de l'étiquetage d'un DISPOSITIF MÉDICAL de l'assemblage d'un SYSTÈME ou de l'adaptation d'un DISPOSITIF MÉDICAL avant mise sur le marché et/ou mise en service indépendamment du fait que ces opérations soient effectuées par cette personne ou par une tierce partie pour le compte de cette personne

[ISO 14971:2000, définition 2.6]

3.11
DISPOSITIF MÉDICAL
tout instrument, appareil, équipement, machine, dispositif, implant, réactif *in vitro* ou calibreur, logiciel, matériel ou autre article similaire ou associé dont le FABRICANT prévoit qu'il soit utilisé seul ou en association chez l'être humain pour la ou les fins spécifiques suivantes:

– diagnostic, prévention, contrôle, traitement ou atténuation d'une maladie,
– diagnostic, contrôle, traitement, atténuation ou compensation d'une blessure,
– étude, remplacement, modification ou entretien de l'anatomie ou d'un PROCESSUS physiologique,
– entretien (artificiel) ou maintien de la vie,
– maîtrise de la conception,
– désinfection des DISPOSITIFS MÉDICAUX,
– communication d'informations à des fins médicales par un examen *in vitro* de spécimens (prélèvement) provenant du corps humain,

et dont l'action principale voulue dans ou sur le corps humain, n'est pas obtenue par des moyens pharmacologiques ou immunologiques ni par métabolisme mais dont la fonction peut être assistée par de tels moyens

NOTE 1 Cette définition a été élaborée par le Groupe de Travail d'Harmonisation Mondiale. Voir la référence bibliographique [15] (dans l'ISO 13485:2003).

[ISO 13485:2003, définition 3.7]

NOTE 2 Les définitions utilisées dans les réglementations de chaque pays peuvent présenter certaines différences.

3.12
LOGICIEL DE DISPOSITIF MÉDICAL
SYSTÈME LOGICIEL qui a été développé pour être incorporé dans le DISPOSITIF MÉDICAL en cours de développement ou qui est destiné à être utilisé comme un DISPOSITIF MÉDICAL à part entière

3.13
RAPPORT DE PROBLÈME
enregistrement du comportement réel ou potentiel d'un PRODUIT LOGICIEL qu'un utilisateur ou une autre personne concernée considère être non sûr, inadéquat pour l'usage prévu ou contraire aux spécifications

3.8
HARM
physical injury, damage, or both to the health of people or damage to property or the environment

[ISO/IEC Guide 51:1999, definition 3.3]

3.9
HAZARD
potential source of HARM

[ISO/IEC Guide 51:1999, definition 3.5]

3.10
MANUFACTURER
natural or legal person with responsibility for designing, manufacturing, packaging, or labelling a MEDICAL DEVICE; assembling a SYSTEM; or adapting a MEDICAL DEVICE before it is placed on the market and/or put into service, regardless of whether these operations are carried out by that person or by a third party on that person's behalf

[ISO 14971:2000, definition 2.6]

3.11
MEDICAL DEVICE
any instrument, apparatus, implement, machine, appliance, implant, in vitro reagent or calibrator, software, material or other similar or related article, intended by the MANUFACTURER to be used, alone or in combination, for human beings for one or more of the specific purpose(s) of

– diagnosis, prevention, monitoring, treatment or alleviation of disease,
– diagnosis, monitoring, treatment, alleviation of or compensation for an injury,
– investigation, replacement, modification, or support of the anatomy or of a physiological PROCESS,
– supporting or sustaining life,
– control of conception,
– disinfection of MEDICAL DEVICES,
– providing information for medical purposes by means of in vitro examination of specimens derived from the human body,

and which does not achieve its primary intended action in or on the human body by pharmacological, immunological or metabolic means, but which may be assisted in its function by such means

NOTE 1 This definition has been developed by the Global Harmonization Task Force (GHTF). See bibliographic reference [15] (in ISO 13485:2003).

[ISO 13485:2003, definition 3.7]

NOTE 2 Some differences can occur in the definitions used in regulations of each country.

3.12
MEDICAL DEVICE SOFTWARE
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 in its own right

3.13
PROBLEM REPORT
a record of actual or potential behaviour of a SOFTWARE PRODUCT that a user or other interested person believes to be unsafe, inappropriate for the intended use or contrary to specification

NOTE 1 La présente norme n'exige pas que chaque RAPPORT DE PROBLÈME donne lieu à une modification du PRODUIT LOGICIEL. Un FABRICANT peut en effet rejeter un RAPPORT DE PROBLÈME en considérant qu'il s'agit d'un malentendu, d'une erreur ou d'un événement insignifiant.

NOTE 2 Un RAPPORT DE PROBLÈME peut concerner un PRODUIT LOGICIEL diffusé ou encore en cours de développement.

NOTE 3 La présente norme exige que le FABRICANT suive des étapes décisionnelles supplémentaires (voir l'Article 6) pour un RAPPORT DE PROBLÈME relatif à un produit diffusé afin de s'assurer que les mesures réglementaires pertinentes sont correctement identifiées et mises en œuvre.

3.14
PROCESSUS
ensemble D'ACTIVITÉS corrélées ou interactives qui transforme des éléments d'entrée en éléments de sortie

[ISO 9000:2000, définition 3.4.1]

NOTE Le terme «ACTIVITÉS» couvre l'utilisation des ressources.

3.15
ESSAI DE RÉGRESSION
essai exigé pour s'assurer qu'un changement d'un composant du SYSTÈME n'a pas altéré la fonctionnalité, la fiabilité ou les performances et n'a pas entraîné de défauts supplémentaires

[ISO/CEI 90003:2004, définition 3.11]

3.16
RISQUE
combinaison de la probabilité d'un DOMMAGE et de sa gravité

[ISO/CEI Guide 51:1999, définition 3.2]

3.17
ANALYSE DU RISQUE
utilisation des informations disponibles pour identifier les PHÉNOMÈNES DANGEREUX (DANGERS) et estimer le RISQUE

[ISO/CEI Guide 51:1999, définition 3.10]

3.18
MAÎTRISE DU RISQUE
PROCESSUS au cours duquel des décisions sont prises et des RISQUEs sont réduits ou maintenus à des niveaux spécifiés

[ISO 14971:2000, définition 2.16, modifiée]

3.19
GESTION DES RISQUES
application systématique de politiques, de procédures et de pratiques de gestion aux TÂCHES d'analyse, d'évaluation et de maîtrise du RISQUE

[ISO 14971:2000, définition 2.18]

3.20
DOSSIER DE GESTION DES RISQUES
ensemble d'enregistrements et autres documents qui ne sont pas nécessairement contigus et qui sont produits par un PROCESSUS DE GESTION DES RISQUES

[ISO 14971:2000, définition 2.19]

NOTE 1 This standard does not require that every PROBLEM REPORT results in a change to the SOFTWARE PRODUCT. A MANUFACTURER can reject a PROBLEM REPORT as a misunderstanding, error or insignificant event.

NOTE 2 A PROBLEM REPORT can relate to a released SOFTWARE PRODUCT or to a SOFTWARE PRODUCT that is still under development.

NOTE 3 This standard requires the MANUFACTURER to perform extra decision making steps (see Clause 6) for a PROBLEM REPORT relating to a released product to ensure that regulatory actions are identified and implemented.

3.14
PROCESS
a set of interrelated or interacting ACTIVITIES that transform inputs into outputs

[ISO 9000:2000, definition 3.4.1]

NOTE The term "ACTIVITIES" covers use of resources.

3.15
REGRESSION TESTING
the testing required to determine that a change to a SYSTEM component has not adversely affected functionality, reliability or performance and has not introduced additional defects

[ISO/IEC 90003:2004, definition 3.11]

3.16
RISK
combination of the probability of occurrence of HARM and the severity of that HARM

[ISO/IEC Guide 51:1999 definition 3.2]

3.17
RISK ANALYSIS
systematic use of available information to identify HAZARDS and to estimate the RISK

[ISO/IEC Guide 51:1999 definition 3.10]

3.18
RISK CONTROL
PROCESS in which decisions are made and RISKS are reduced to, or maintained within, specified levels

[ISO 14971:2000 definition 2.16, modified]

3.19
RISK MANAGEMENT
systematic application of management policies, procedures, and practices to the TASKS of analyzing, evaluating, and controlling RISK

[ISO 14971:2000 definition 2.18]

3.20
RISK MANAGEMENT FILE
set of records and other documents, not necessarily contiguous, that are produced by a RISK MANAGEMENT PROCESS

[ISO 14971:2000 definition 2.19]

3.21
SÉCURITÉ
absence de RISQUE inacceptable

[ISO/CEI Guide 51:1999, définition 3.1]

3.22
SÛRETÉ
protection des informations et des données de sorte que des personnes ou des SYSTÈMES non autorisés ne puissent les lire ou les modifier et que l'accès à ces informations et données ne soit pas refusé à des personnes ou des SYSTÈMES autorisés

[ISO/CEI 12207:1995, définition 3.25]

3.23
BLESSURE GRAVE
blessure ou maladie qui, directement ou indirectement:

a) menace la vie,

b) entraîne une carence permanente d'une fonction physiologique ou endommage de manière définitive une structure du corps, ou

c) nécessite une intervention médicale ou chirurgicale pour prévenir une carence permanente d'une fonction physiologique ou un endommagement définitif d'une structure du corps

NOTE Carence permanente signifie une carence irréversible ou un endommagement d'une structure du corps ou d'une fonction, à l'exclusion des carences ou préjudices insignifiants.

3.24
MODÈLE DU CYCLE DE VIE DE DÉVELOPPEMENT DU LOGICIEL
structure conceptuelle couvrant la vie du logiciel depuis la définition de ses exigences jusqu'à sa mise en fabrication et qui:

– identifie le PROCESSUS, les ACTIVITÉS et TÂCHES impliqués dans le développement d'un PRODUIT LOGICIEL,

– décrit l'ordre et la dépendance entre ACTIVITÉS et TÂCHES, et

– identifie les repères auxquels la complétude des LIVRABLES spécifiés est vérifiée.

NOTE Basée sur la définition 3.11 de l'ISO/CEI 12207:1995

3.25
ÉLÉMENT LOGICIEL
toute partie identifiable d'un programme informatique

[ISO/CEI 90003:2004, définition 3.14, modifiée]

NOTE Trois termes identifient la décomposition du logiciel. Le niveau supérieur est le SYSTÈME LOGICIEL. Le niveau le plus bas qui n'est pas décomposé plus en avant est l'UNITÉ LOGICIELLE. Tous les niveaux de composition, y compris les niveaux supérieur et inférieur, peuvent être dénommés ÉLÉMENTS LOGICIEL. Un SYSTÈME LOGICIEL est donc composé d'un ou de plusieurs ÉLÉMENTS LOGICIEL, et chaque ÉLÉMENT LOGICIEL est composé d'une ou de plusieurs UNITÉS LOGICIELLES ou d'un ou de plusieurs ÉLÉMENTS LOGICIELS décomposables. Il incombe au FABRICANT de fournir la définition et la granularité des ÉLÉMENTS LOGICIELS et des UNITÉS LOGICIELLES.

3.26
PRODUIT LOGICIEL
ensemble constitué de programmes informatiques, de procédures, et des données et documentation éventuellement associées

[ISO/CEI 12207:1995, définition 3.26]

3.27
SYSTÈME LOGICIEL
ensemble intégré d'ÉLÉMENTS LOGICIELS organisé de manière à réaliser une fonction ou un ensemble de fonctions spécifiques

3.21
SAFETY
freedom from unacceptable RISK

[ISO/IEC Guide 51:1999 definition 3.1]

3.22
SECURITY
protection of information and data so that unauthorized people or SYSTEMS cannot read or modify them and so that authorized persons or SYSTEMS are not denied access to them

[ISO/IEC 12207:1995 definition 3.25]

3.23
SERIOUS INJURY
injury or illness that directly or indirectly:

a) is life threatening,

b) results in permanent impairment of a body function or permanent damage to a body structure, or

c) necessitates medical or surgical intervention to prevent permanent impairment of a body function or permanent damage to a body structure

NOTE Permanent impairment means an irreversible impairment or damage to a body structure or function excluding trivial impairment or damage.

3.24
SOFTWARE DEVELOPMENT LIFE CYCLE MODEL
conceptual structure spanning the life of the software from definition of its requirements to its release for manufacturing, which:

– identifies the PROCESS, ACTIVITIES and TASKS involved in development of a SOFTWARE PRODUCT,

– describes the sequence of and dependency between ACTIVITIES and TASKS, and

– identifies the milestones at which the completeness of specified DELIVERABLES is verified.

NOTE Based on ISO/IEC 12207:1995, definition 3.11

3.25
SOFTWARE ITEM
any identifiable part of a computer program

[ISO/IEC 90003:2004, definition 3.14, modified]

NOTE 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 definition and granularity of the SOFTWARE ITEMS and SOFTWARE UNITS.

3.26
SOFTWARE PRODUCT
set of computer programs, procedures, and possibly associated documentation and data

[ISO/IEC 12207:1995 definition 3.26]

3.27
SOFTWARE SYSTEM
integrated collection of SOFTWARE ITEMS organized to accomplish a specific function or set of functions

3.28
UNITÉ LOGICIELLE
ÉLÉMENT LOGICIEL qui n'est pas subdivisé en d'autres éléments

NOTE Les UNITÉS LOGICIELLES peuvent être utilisées pour essais ou gestion de la configuration du logiciel.

3.29
SOUP (sigle pour l'anglais «Software Of Unknown Provenance»)
logiciel de provenance inconnue
ÉLÉMENT LOGICIEL qui est déjà développé, et généralement disponible, et qui n'a pas été développé pour être incorporé dans le DISPOSITIF MÉDICAL (également appelé «logiciel de série») ou logiciel précédemment développé pour lequel les enregistrements suffisants des processus de développement ne sont pas disponibles

3.30
SYSTÈME
ensemble composite intégré constitué d'un ou de plusieurs PROCESSUS, matériels, logiciels, fonctionnalités et individus qui fournissent une aptitude à satisfaire un besoin ou un objectif déclaré

[ISO/CEI 12207:1995, définition 3.31]

3.31
TÂCHE
partie unique d'un travail qui doit être effectué

3.32
TRAÇABILITÉ
degré auquel une relation peut être établie entre deux ou plusieurs produits du PROCESSUS de développement

[IEEE 610.12:1990]

3.33
VÉRIFICATION
confirmation par des preuves tangibles que les exigences spécifiées ont été satisfaites

NOTE 1 Le terme «vérifié» désigne l'état correspondant.

[ISO 9000:2000, définition 3.8.4]

NOTE 2 En conception et développement, la VÉRIFICATION est le PROCESSUS d'examen du résultat d'une ACTIVITÉ donnée afin de déterminer la conformité à la prescription définie pour ladite ACTIVITÉ.

3.34
VERSION
instance identifiée d'un ÉLÉMENT DE CONFIGURATION

NOTE 1 La modification d'une VERSION d'un PRODUIT LOGICIEL, donnant lieu à une nouvelle VERSION exige une action de gestion de la configuration du logiciel.

NOTE 2 Basé sur [ISO/CEI 12207:1995, définition 3.37]

4 * Exigences générales

4.1 * Système de management de la qualité

Le FABRICANT du LOGICIEL DE DISPOSITIF MÉDICAL doit démontrer la capacité à fournir un LOGICIEL DE DISPOSITIF MÉDICAL qui réponde de manière cohérente aux exigences du client et aux exigences réglementaires applicables.

3.28
SOFTWARE UNIT
SOFTWARE ITEM that is not subdivided into other items

NOTE SOFTWARE UNITS can be used for the purpose of software configuration management or testing.

3.29
SOUP
software of unknown provenance (acronym)
SOFTWARE ITEM that is already developed and generally available and that has not been developed for the purpose of being incorporated into the MEDICAL DEVICE (also known as "off-the-shelf software") or software previously developed for which adequate records of the development PROCESSES are not available

3.30
SYSTEM
integrated composite consisting of one or more of the PROCESSES, hardware, software, facilities, and people, that provides a capability to satisfy a stated need or objective

[ISO/IEC 12207:1995, definition 3.31]

3.31
TASK
a single piece of work that needs to be done

3.32
TRACEABILITY
degree to which a relationship can be established between two or more products of the development PROCESS

[IEEE 610.12:1990]

3.33
VERIFICATION
confirmation through provision of objective evidence that specified requirements have been fulfilled

NOTE 1 "Verified" is used to designate the corresponding status.

[ISO 9000:2000, definition 3.8.4]

NOTE 2 In design and development, VERIFICATION concerns the PROCESS of examining the result of a given ACTIVITY to determine conformity with the stated requirement for that ACTIVITY.

3.34
VERSION
identified instance of a CONFIGURATION ITEM

NOTE 1 Modification to a VERSION of a SOFTWARE PRODUCT, resulting in a new VERSION, requires software configuration management action.

NOTE 2 Based on ISO/IEC 12207:1995, definition 3.37.

4 * General requirements

4.1 * Quality management system

The MANUFACTURER of MEDICAL DEVICE SOFTWARE shall demonstrate the ability to provide MEDICAL DEVICE SOFTWARE that consistently meets customer requirements and applicable regulatory requirements.

NOTE 1 La démonstration de cette capacité peut se faire par l'utilisation d'un système de management de la qualité conforme à:

– l'ISO 13485 [7] ; ou

– une norme nationale de système de management de la qualité ; ou

– un système de management de la qualité exigé par une réglementation nationale.

NOTE 2 L'ISO/CEI 90003 [11] fournit des lignes directrices pour l'application des exigences d'un système de management de la qualité au logiciel.

4.2 * GESTION DES RISQUES

Le FABRICANT doit appliquer un PROCESSUS DE GESTION DES RISQUES conforme à l'ISO 14971.

4.3 * Classification de sécurité du logiciel

a) Le FABRICANT doit attribuer à chaque SYSTÈME LOGICIEL une classe de SÉCURITÉ du logiciel (A, B ou C) en fonction des effets possibles sur le patient, l'opérateur ou d'autres personnes résultant d'un PHÉNOMÈNE DANGEREUX auquel le SYSTÈME LOGICIEL peut contribuer.

   Les classes de SÉCURITÉ du logiciel doivent au départ être attribuées en se basant sur le degré de sévérité suivant:

   Classe A : Aucune blessure ou atteinte à la santé n'est possible

   Classe B : Une BLESSURE NON GRAVE est possible

   Classe C : La mort ou une BLESSURE GRAVE est possible

   Si le PHÉNOMÈNE DANGEREUX peut résulter d'une défaillance du SYSTÈME LOGICIEL à se comporter conformément aux spécifications, la probabilité d'une telle défaillance doit être supposée de 100 %.

   Si le RISQUE de mort ou de BLESSURE GRAVE provenant d'une défaillance logicielle est ensuite ramené à un niveau acceptable (tel que défini par l'ISO 14971) par des mesures de MAÎTRISE DES RISQUES matérielles, soit en réduisant les conséquences de la défaillance, soit en réduisant la probabilité de mort ou de BLESSURE GRAVE provenant de cette défaillance, la classe de sécurité du logiciel peut être ramenée de C à B. Si le RISQUE de BLESSURE non GRAVE provenant d'une défaillance logicielle est de la même manière ramené à un niveau acceptable par des mesures de MAÎTRISE DES RISQUES matérielles, la classe de sécurité du logiciel peut être ramenée de B à A.

b) Le FABRICANT doit attribuer à chaque SYSTÈME LOGICIEL qui contribue à la mise en œuvre des mesures de MAÎTRISE DES RISQUES une classe de sécurité du logiciel, fondée sur les effets possibles du PHÉNOMÈNE DANGEREUX qui sont couverts par la mesure de MAÎTRISE DU RISQUE.

c) Le FABRICANT doit consigner la classe de sécurité du logiciel attribuée à chaque SYSTÈME LOGICIEL dans le dossier de GESTION DES RISQUES.

d) Lorsqu'un SYSTÈME LOGICIEL est décomposé en ÉLÉMENTS LOGICIELS et qu'un ÉLÉMENT LOGICIEL est décomposé en d'autres ÉLÉMENTS LOGICIELS, ces ÉLÉMENTS LOGICIELs doivent hériter de la classe de sécurité du logiciel de l'ÉLÉMENT LOGICIEL initial (ou du SYSTÈME LOGICIEL) à moins que le FABRICANT ne justifie par écrit une classification dans une classe différente de sécurité du logiciel. Une telle justification doit expliquer la manière dont les nouveaux ÉLÉMENTS LOGICIELS sont différenciés pour pouvoir être classés séparément.

e) Le FABRICANT doit consigner la classe de sécurité du logiciel de chaque élément LOGICIEL si cette classe est différente de la classe de l'ÉLÉMENT LOGICIEL à partir duquel il a été créé par décomposition.

f) Pour la conformité à la présente norme, lorsqu'un PROCESSUS est exigé pour les ÉLÉMENTS LOGICIELS d'une classe spécifique et que ce PROCESSUS est nécessairement appliqué à un groupe d'ÉLÉMENTS LOGICIELS, le FABRICANT doit utiliser les PROCESSUS et TÂCHES qui sont exigés par la classe de sécurité de l'ÉLÉMENT LOGICIEL la plus élevée définie dans le groupe à moins que le FABRICANT ne justifie par écrit dans le dossier de GESTION DES RISQUES, l'utilisation d'une classification plus basse.

NOTE 1 Demonstration of this ability can be by the use of a quality management system that complies with:

- ISO 13485 [7]; or

- a national quality management system standard; or

- a quality management system required by national regulation.

NOTE 2 Guidance for applying quality management system requirements to software can be found in ISO/IEC 90003 [11].

4.2 * RISK MANAGEMENT

The MANUFACTURER shall apply a RISK MANAGEMENT PROCESS complying with ISO 14971.

4.3 * Software safety classification

a) The MANUFACTURER shall assign to each SOFTWARE SYSTEM a software safety class (A, B, or C) according to the possible effects on the patient, operator, or other people resulting from a HAZARD to which the SOFTWARE SYSTEM can contribute.

   The software safety classes shall initially be assigned based on severity as follows:

   Class A: No injury or damage to health is possible

   Class B: Non-SERIOUS INJURY is possible

   Class C: Death or SERIOUS INJURY is possible

   If the HAZARD could arise from a failure of the SOFTWARE SYSTEM to behave as specified, the probability of such failure shall be assumed to be 100 percent.

   If the RISK of death or SERIOUS INJURY arising from a software failure is subsequently reduced to an acceptable level (as defined by ISO 14971) by a hardware RISK CONTROL measure, either by reducing the consequences of the failure or by reducing the probability of death or SERIOUS INJURY arising from that failure, the software safety classification may be reduced from C to B; and if the RISK of non-SERIOUS INJURY arising from a software failure is similarly reduced to an acceptable level by a hardware RISK CONTROL measure, the software safety classification may be reduced from B to A.

b) The MANUFACTURER shall assign to each SOFTWARE SYSTEM that contributes to the implementation of a RISK CONTROL measure a software safety class based on the possible effects of the HAZARD that the RISK CONTROL measure is controlling.

c) The MANUFACTURER shall document the software safety class assigned to each SOFTWARE SYSTEM in the RISK MANAGEMENT FILE.

d) When a SOFTWARE SYSTEM is decomposed into SOFTWARE ITEMS, and when a SOFTWARE ITEM is decomposed into further SOFTWARE ITEMS, such SOFTWARE ITEMS shall inherit the software safety classification of the original SOFTWARE ITEM (or SOFTWARE SYSTEM) unless the MANUFACTURER documents a rationale for classification into a different software safety class. Such a rationale shall explain how the new SOFTWARE ITEMS are segregated so that they may be classified separately.

e) The MANUFACTURER shall document the software safety class of each SOFTWARE ITEM if that class is different from the class of the SOFTWARE ITEM from which it was created by decomposition.

f) For compliance with this standard, wherever a PROCESS is required for SOFTWARE ITEMS of a specific classification and the PROCESS is necessarily applied 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.

g) Pour chaque SYSTÈME LOGICIEL, les exigences de la classe C doivent être appliquées jusqu'à attribution d'une classe de SÉCURITÉ du logiciel.

NOTE Dans les exigences qui suivent, les classes de sécurité du logiciel pour lesquelles l'exigence doit être réalisée sont identifiées en suivant l'exigence sous la forme [Classe...].

5 PROCESSUS de développement du logiciel

5.1 * Planification du développement du logiciel

5.1.1 Plan de développement du logiciel

Le FABRICANT doit établir un(des) plan(s) de développement du logiciel pour entreprendre des ACTIVITÉS DU PROCESSUS DE DÉVELOPPEMENT DU LOGICIEL convenant au domaine d'application, à l'importance et aux classes de sécurité du logiciel du SYSTÈME LOGICIEL à développer. Le MODÈLE DE CYCLE DE VIE DE DÉVELOPPEMENT DU LOGICIEL doit être soit complètement défini, soit référencé dans le ou les plans. Le plan doit traiter des éléments suivants:

a) les PROCESSUS à utiliser au cours du développement du SYSTÈME LOGICIEL (voir Note 4);

b) les LIVRABLES (y compris la documentation) des ACTIVITÉS et TÂCHES;

c) la TRAÇABILITÉ entre les exigences du SYSTÈME, les exigences du logiciel, les essais du SYSTÈME LOGICIEL et les mesures de MAÎTRISE DU RISQUE mis en œuvre dans le logiciel;

d) la gestion de la configuration et des modifications du logiciel, y compris les ÉLÉMENTS DE CONFIGURATION de LOGICIEL DE PROVENANCE INCONNUE (SOUP) utilisés à l'appui du développement; et

e) la résolution des problèmes de logiciel pour le traitement des problèmes détectés dans les PRODUITS LOGICIELS, dans les LIVRABLES et dans les ACTIVITÉS à chaque étape du cycle de vie.

[Classes A, B, C]

NOTE 1 LE MODÈLE DU CYCLE DE VIE DE DÉVELOPPEMENT DU LOGICIEL peut identifier différents éléments (PROCESSUS, ACTIVITÉS, TÂCHES et LIVRABLES) pour différents ÉLÉMENTS LOGICIELS en fonction de la classe de sécurité du logiciel de chaque L'ÉLÉMENT LOGICIEL du SYSTÈME LOGICIEL.

NOTE 2 Il est possible que ces ACTIVITÉS et TÂCHES se chevauchent ou interagissent et elles peuvent être réalisées de manière itérative ou récursive. La présente norme n'a pas pour but de recommander l'utilisation d'un modèle de cycle de vie spécifique.

NOTE 3 D'autres PROCESSUS sont décrits dans la présente norme indépendamment du PROCESSUS de développement. Ceci ne signifie pas qu'ils doivent être mis en œuvre comme des ACTIVITÉS et TÂCHES séparées. Les ACTIVITÉS et TÂCHES des autres PROCESSUS peuvent être intégrées dans le PROCESSUS de développement.

NOTE 4 Le plan de développement du logiciel peut référencer des PROCESSUS existants ou en définir de nouveaux.

NOTE 5 Le plan de développement du logiciel peut être intégré dans un plan de développement global du SYSTÈME.

5.1.2 Mise à jour du plan de développement logiciel

Le FABRICANT doit mettre à jour le plan au fur et à mesure du développement, le cas échéant. [Classes A, B, C]

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

a) Le FABRICANT doit référencer les exigences du SYSTÈME en tant qu'éléments d'entrée dans le plan de développement du logiciel.

b) Le FABRICANT doit intégrer ou référencer dans le plan de développement du logiciel les procédures de coordination du développement du logiciel ainsi que de validation de la conception et du développement qui sont nécessaires POUR satisfaire aux exigences du 4.1.

[Classes A, B, C]

g) For each SOFTWARE SYSTEM, until a software safety class is assigned, Class C requirements shall apply.

NOTE In the requirements that follow, the software safety classes that the requirement must be performed for are identified following the requirement in the form [Class . . .].

5 Software development PROCESS

5.1 * Software development planning

5.1.1 Software development plan

The MANUFACTURER shall establish a software development plan (or plans) for conducting the ACTIVITIES of the software development PROCESS appropriate to the scope, magnitude, and software safety classifications of the SOFTWARE SYSTEM to be developed. The SOFTWARE DEVELOPMENT LIFE CYCLE MODEL shall either be fully defined or be referenced in the plan (or plans). The plan shall address the following:

a) the PROCESSES to be used in the development of the SOFTWARE SYSTEM (see Note 4);

b) the DELIVERABLES (includes documentation) of the ACTIVITIES and TASKS;

c) TRACEABILITY between SYSTEM requirements, software requirements, SOFTWARE SYSTEM test, and RISK CONTROL measures implemented in software;

d) software configuration and change management, including SOUP CONFIGURATION ITEMS and software used to support development; and

e) software problem resolution for handling problems detected in the SOFTWARE PRODUCTS, DELIVERABLES and ACTIVITIES at each stage of the life cycle.

[Class A, B, C]

NOTE 1 The SOFTWARE DEVELOPMENT LIFE CYCLE MODEL can identify different elements (PROCESSES, ACTIVITIES, TASKS and DELIVERABLES) for different SOFTWARE ITEMS according to the software safety classification of each SOFTWARE ITEM of the SOFTWARE SYSTEM.

NOTE 2 These ACTIVITIES and TASKS can overlap or interact and can be performed iteratively or recursively. It is not the intent to imply that a specific life cycle model should be used.

NOTE 3 Other PROCESSES are described in this standard separately from the development PROCESS. This does not imply that they must be implemented as separate ACTIVITIES and TASKS. The ACTIVITIES and TASKS of the other PROCESSES can be integrated into the development PROCESS.

NOTE 4 The software development plan can reference existing PROCESSES or define new ones.

NOTE 5 The software development plan may be integrated in an overall SYSTEM development plan.

5.1.2 Keep software development plan updated

The MANUFACTURER shall update the plan as development proceeds as appropriate. [Class A, B, C]

5.1.3 Software development plan reference to SYSTEM design and development

a) As inputs for software development, SYSTEM requirements shall be referenced in the software development plan by the MANUFACTURER.

b) The MANUFACTURER shall include or reference in the software development plan procedures for coordinating the software development and the design and development validation necessary to satisfy 4.1.

[Class A, B, C]

NOTE Il pourrait ne pas y avoir de différence entre les exigences du SYSTÈME LOGICIEL et les exigences du SYSTÈME si le SYSTÈME LOGICIEL est un SYSTÈME autonome (dispositif uniquement logiciel).

5.1.4 Planification des normes, méthodes et outils de développement du logiciel

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

a) les normes,

b) les méthodes et

c) les outils

associés au développement des ÉLÉMENTS LOGICIELS de classe C. [Classe C]

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

Le FABRICANT doit inclure ou référencer dans le plan de développement du logiciel un plan d'intégration des ÉLÉMENTS LOGICIELS (y compris les logiciels SOUP) et de réalisation d'essais pendant l'intégration. [Classes B, C]

NOTE Il est admis de combiner en un seul plan et en un seul ensemble d'ACTIVITÉS, les essais d'intégration et les essais du SYSTÈME LOGICIEL.

5.1.6 Planification de la VÉRIFICATION du logiciel

Le FABRICANT doit inclure ou référencer dans le plan de développement du logiciel les informations suivantes relatives à la VÉRIFICATION:

a) LES LIVRABLES qui nécessitent une VÉRIFICATION;

b) les TÂCHEs de VÉRIFICATION requises pour chaque ACTIVITÉ du cycle de vie;

c) les étapes auxquelles les LIVRABLES sont VÉRIFIÉS; et

d) les critères d'acceptation de la VÉRIFICATION des LIVRABLES.

[Classes A, B, C]

5.1.7 Planification de la GESTION DES RISQUES du logiciel

Le FABRICANT doit inclure ou référencer dans le plan de développement du logiciel, un plan relatif à la réalisation des ACTIVITÉS et TÂCHES du PROCESSUS DE GESTION DES RISQUES du logiciel, y compris la gestion des RISQUEs liés aux logicielS SOUP. [Classes A, B, C]

NOTE Voir Article 7.

5.1.8 Planification de la documentation

Le FABRICANT doit inclure ou référencer dans le plan de développement du logiciel, les informations relatives aux documents à produire pendant le cycle de vie de développement du logiciel. Pour chaque document identifié ou type de document, les informations suivantes doivent être incluses ou référencées:

a) le titre, le nom ou la convention de désignation;

b) l'objet;

c) l'audience/la diffusion à laquelle le document est destiné; et

d) les procédures et responsabilités de développement, de revue, d'approbation et de modification.

[Classes A, B, C]

NOTE There might not be a difference between SOFTWARE SYSTEM requirements and SYSTEM requirements if the SOFTWARE SYSTEM is a stand alone SYSTEM (software-only device).

5.1.4 Software development standards, methods and tools planning

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

a) standards,

b) methods, and

c) tools

associated with the development of SOFTWARE ITEMS of class C. [Class C]

5.1.5 Software integration and integration testing planning

The MANUFACTURER shall include or reference in the software development plan, a plan to integrate the SOFTWARE ITEMS (including SOUP) and perform testing during integration. [Class B, C]

NOTE It is acceptable to combine integration testing and SOFTWARE SYSTEM testing into a single plan and set of ACTIVITIES.

5.1.6 Software VERIFICATION planning

The MANUFACTURER shall include or reference in the software development plan the following VERIFICATION information:

a) DELIVERABLES requiring VERIFICATION;

b) the required VERIFICATION TASKS for each life cycle ACTIVITY;

c) milestones at which the DELIVERABLES are VERIFIED; and

d) the acceptance criteria for VERIFICATION of the DELIVERABLES.

[Class A, B, C]

5.1.7 Software RISK MANAGEMENT planning

The MANUFACTURER shall include or reference in the software development plan, a plan to conduct the ACTIVITIES and TASKS of the software RISK MANAGEMENT PROCESS, including the management of RISKS relating to SOUP. [Class A, B, C]

NOTE See Clause 7.

5.1.8 Documentation planning

The MANUFACTURER shall include or reference in the software development plan information about the documents to be produced during the software development life cycle. For each identified document or type of document the following information shall be included or referenced:

a) title, name or naming convention;

b) purpose;

c) intended audience of document; and

d) procedures and responsibilities for development, review, approval and modification.

[Class A, B, C]

5.1.9 Planification de la gestion de configuration du logiciel

Le FABRICANT doit inclure ou référencer les informations relatives à la gestion de la configuration du logiciel dans le plan de développement du logiciel. Les informations de gestion de la configuration du logiciel doivent comprendre ou référencer:

a) les classes, types, catégories ou listes d'éléments à contrôler;

b) les ACTIVITÉs et TÂCHES de gestion de la configuration du logiciel;

c) la ou les organisation(s) chargée(s) de réaliser la gestion de la configuration du logiciel et les ACTIVITÉs correspondantes;

d) leur lien avec d'autres organisations telles que celles chargées du développement ou de la maintenance du logiciel;

e) le moment où les éléments doivent être mis sous contrôle de la configuration; et

f) le moment où le PROCESSUS de résolution du problème doit être utilisé.

[Classes A, B, C]

5.1.10 Eléments annexes à contrôler

Les éléments à contrôler doivent inclure les outils, les éléments ou les réglages utilisés pour développer le LOGICIEL DE DISPOSITIF MÉDICAL, qui pourraient avoir un impact sur le LOGICIEL DE DISPOSITIF MÉDICAL. [Classes B, C]

NOTE De tels éléments comprennent par exemple les versions de programmes de compilation/ de langages d'assemblage, les fichiers makefile, les fichiers séquentiels et les réglages environnementaux spécifiques.

5.1.11 Contrôle de l'ÉLÉMENT DE CONFIGURATION du logiciel avant VÉRIFICATION

Le FABRICANT doit planifier la mise des ÉLÉMENTS DE CONFIGURATION sous contrôle documenté de la gestion de la configuration avant qu'ils ne soient VÉRIFIÉS. [Classes B, C]

5.2 * Analyses des exigences du logiciel

5.2.1 Définition et documentation des exigences du logiciel d'après les exigences du SYSTÈME

Pour chaque SYSTÈME LOGICIEL du DISPOSITIF MÉDICAL, le FABRICANT doit définir et consigner les exigences du SYSTÈME LOGICIEL à partir des exigences au niveau du SYSTÈME. [Classes A, B, C]

NOTE Il pourrait ne pas y avoir de différence entre les exigences du SYSTÈME LOGICIEL et les exigences du SYSTÈME si le SYSTÈME LOGICIEL est un SYSTÈME autonome (dispositif uniquement logiciel).

5.2.2 Teneur des exigences du logiciel

Selon la pertinence pour le LOGICIEL DU DISPOSITIF MÉDICAL, le FABRICANT doit inclure dans les exigences du logiciel:

a) les exigences en termes de fonctionnalité et de capacité;

NOTE 1 Les exemples comprennent:

– la performance (par exemple, objet du logiciel, exigences de synchronisation),

– les caractéristiques physiques (par exemple, langage de codage, plate-forme, système d'exploitation),

– l'environnement informatique (par exemple matériel, taille de la mémoire, unité centrale, zone de temps, synchronisation, infrastructure du réseau) dans lequel le logiciel doit fonctionner, et

– le besoin de compatibilité avec des mises à niveau ou des versions multiples de logiciel de provenance inconnue (SOUP) ou d'autres dispositifs.

5.1.9 Software configuration management planning

The MANUFACTURER shall include or reference software configuration management information in the software development plan. The software configuration management information shall include or reference:

a) the classes, types, categories or lists of items to be controlled;

b) the software configuration management ACTIVITIES and TASKS;

c) the organization(s) responsible for performing software configuration management and ACTIVITIES;

d) their relationship with other organizations, such as software development or maintenance;

e) when the items are to be placed under configuration control; and

f) when the problem resolution PROCESS is to be used.

[Class A, B, C]

5.1.10 Supporting items to be controlled

The items to be controlled shall include tools, items or settings, used to develop the MEDICAL DEVICE SOFTWARE, which could impact the MEDICAL DEVICE SOFTWARE. [Class B, C]

NOTE Examples of such items include compiler/assembler versions, make files, batch files, and specific environment settings.

5.1.11 Software CONFIGURATION ITEM control before VERIFICATION

The MANUFACTURER shall plan to place CONFIGURATION ITEMS under documented configuration management control before they are VERIFIED. [Class B, C]

5.2 * Software requirements analysis

5.2.1 Define and document software requirements from SYSTEM requirements

For each SOFTWARE SYSTEM of the MEDICAL DEVICE, the MANUFACTURER shall define and document SOFTWARE SYSTEM requirements from the SYSTEM level requirements. [Class A, B, C]

NOTE There might not be a difference between SOFTWARE SYSTEM requirements and SYSTEM requirements if the SOFTWARE SYSTEM is a stand alone SYSTEM (software-only device).

5.2.2 Software requirements content

As appropriate to the MEDICAL DEVICE SOFTWARE, the MANUFACTURER shall include in the software requirements:

a) functional and capability requirements;

NOTE 1 Examples include:

– performance (e.g., purpose of software, timing requirements),

– physical characteristics (e.g., code language, platform, operating system),

– computing environment (e.g., hardware, memory size, processing unit, time zone, network infrastructure) under which the software is to perform, and

– need for compatibility with upgrades or multiple SOUP or other device versions.

b) Les éléments d'entrée et de sortie DU SYSTÈME LOGICIEL;

NOTE 2 Les exemples comprennent:

– les caractéristiques des données (par exemple, numérique, alphanumérique, format),
– les plages,
– les limites, et
– les valeurs par défaut;

c) Les interfaces entre le SYSTÈME LOGICIEL et d'autres SYSTÈMES;

d) Les alarmes, avertissements et messages opérateurs générés par le logiciel;

e) Les exigences en matière de SÛRETÉ;

NOTE 3 Les exemples comprennent:

– les exigences liées au compromis en matière d'informations sensibles,
– les exigences d'authentification,
– les exigences d'autorisation,
– les exigences d'enregistrement d'audit, et
– les exigences d'intégrité des communications

f) Les exigences de l'ingénierie de l'aptitude à l'utilisation qui sont sensibles aux erreurs humaines et à la formation:

NOTE 4 Les exemples comprennent les exigences liées à:

– l'assistance pour les opérations manuelles,
– les interactions homme-machine,
– les contraintes pour le personnel, et
– les domaines nécessitant une concentration de la part de l'opérateur;

NOTE 5 Les informations relatives aux exigences de l'ingénierie de l'aptitude à l'utilisation sont données dans la CEI 60601-1-6.

g) Les exigences en matière de base de données et de définitions des données;

NOTE 6 Les exemples comprennent:

– le format,
– l'adéquation,
– la fonction.

h) Les exigences d'installation et d'acceptation du logiciel de DISPOSITIF MÉDICAL livré au(x) site(s) d'exploitation et de maintenance;

i) Les exigences liées aux méthodes d'exploitation et de maintenance;

j) La documentation utilisateur à élaborer;

k) Les exigences de maintenance par l'utilisateur; et

l) Les exigences réglementaires.

[Classes A, B, C]

NOTE 7 Il est admis que toutes ces exigences ne soient pas disponibles au début du PROCESSUS de développement du logiciel.

NOTE 8 L'ISO/CEI 9126-1 [8] fournit des informations sur les caractéristiques de la qualité qui peuvent être utiles pour la définition des exigences logicielles.

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

Le FABRICANT doit inclure dans les exigences les mesures de MAÎTRISE DU RISQUE mises en œuvre dans le logiciel pour tenir compte des défaillances matérielles et des éventuels défauts du logiciel, en fonction du DISPOSITIF MÉDICAL. [Classes B, C]

NOTE Il est admis que ces exigences ne soient pas disponibles au début du PROCESSUS de développement du logiciel et peuvent changer au fur et à mesure de la conception du logiciel et de la définition des mesures de MAÎTRISE DU RISQUE.

b) SOFTWARE SYSTEM inputs and outputs;

NOTE 2 Examples include:

– data characteristics (e.g., numerical, alpha-numeric, format)
– ranges,
– limits, and
– defaults.

c) interfaces between the SOFTWARE SYSTEM and other SYSTEMS;

d) software-driven alarms, warnings, and operator messages;

e) SECURITY requirements;

NOTE 3 Examples include:

– those related to the compromise of sensitive information,
– authentication,
– authorization,
– audit trail, and
– communication integrity.

f) usability engineering requirements that are sensitive to human errors and training;

NOTE 4 Examples include those related to:

– support for manual operations,
– human-equipment interactions,
– constraints on personnel, and
– areas needing concentrated human attention.

NOTE 5 Information regarding usability engineering requirements can be found in IEC 60601-1-6.

g) data definition and database requirements;

NOTE 6 Examples include:

– form;
– fit;
– function.

h) installation and acceptance requirements of the delivered MEDICAL DEVICE SOFTWARE at the operation and maintenance site or sites;

i) requirements related to methods of operation and maintenance;

j) user documentation to be developed;

k) user maintenance requirements; and

l) regulatory requirements.

[Class A, B, C]

NOTE 7 All of these requirements might not be available at the beginning of the software development.

NOTE 8 ISO/IEC 9126-1 [8] provides information on quality characteristics that may be useful in defining software requirements.

5.2.3 Include RISK CONTROL measures in software requirements

The MANUFACTURER shall include RISK CONTROL measures implemented in software for hardware failures and potential software defects in the requirements as appropriate to the MEDICAL DEVICE SOFTWARE. [Class B, C]

NOTE These requirements might not be available at the beginning of the software development and can change as the software is designed and RISK CONTROL measures are further defined.

5.2.4 Ré-ÉVALUATION de l'ANALYSE DU RISQUE du DISPOSITIF MÉDICAL

Une fois les exigences du logiciel établies, le FABRICANT doit ré-ÉVALUER et si nécessaire mettre à jour l'ANALYSE DE RISQUE du DISPOSITIF MÉDICAL. [Classes A, B, C]

5.2.5 Mise à jour des exigences du SYSTÈME

Le FABRICANT doit s'assurer que les exigences existantes, y compris les exigences du SYSTÈME sont ré-ÉVALUÉES et correctement mises à jour en fonction des résultats de l'ACTIVITÉ d'analyse des exigences du logiciel. [Classes A, B, C]

5.2.6 Vérification des exigences du logiciel

Le FABRICANT doit vérifier et consigner que les exigences du logiciel:

a) mettent en œuvre les exigences du SYSTÈME, y compris celles liées à la MAÎTRISE DU RISQUE;

b) ne se contredisent pas;

c) sont exprimées en termes univoques:

d) sont indiquées en des termes qui permettent d'établir les critères et les performances des essais de manière à s'assurer que ces critères sont remplis;

e) peuvent être identifiées de manière unique; et

f) leur TRAÇABILITÉ aux exigences du SYSTÈME ou autre source est assurée;

[Classes A, B, C]

NOTE La présente norme n'exige pas le recours à un langage de spécification formel.

5.3 * Conception ARCHITECTURALE du logiciel

5.3.1 Conversion des exigences du logiciel en ARCHITECTURE

Le FABRICANT doit transformer les exigences du LOGICIEL DE DISPOSITIF MÉDICAL en une ARCHITECTURE documentée décrivant la structure du logiciel et identifiant les éléments LOGICIELS. [Classes B, C]

5.3.2 Elaboration d'une ARCHITECTURE pour les interfaces d'ÉLÉMENTS LOGICIELS

Le FABRICANT doit élaborer et documenter une ARCHITECTURE pour les interfaces entre les ÉLÉMENTS LOGICIELS et les composants externes aux ÉLÉMENTS LOGICIELS (tant logiciels que matériels), ainsi qu'entre les ÉLÉMENTS LOGICIELS proprement dits. [Classes B, C]

5.3.3 Spécification des exigences fonctionnelles et de performance des ÉLÉMENTS LOGICIELS SOUP

Si un ÉLÉMENT LOGICIEL est identifié comme étant SOUP, le FABRICANT doit spécifier les exigences fonctionnelles et de performance dudit élément SOUP qui sont nécessaires à son usage prévu. [Classes B, C]

5.3.4 Spécification des matériels et des logiciels SYSTÈME nécessaires à l'ÉLÉMENT LOGICIEL SOUP

Si uN ÉLÉMENT LOGICIEL est identifié comme étant SOUP, le FABRICANT doit spécifier les matériels et logiciels SYSTÈME nécessaires pour assurer le fonctionnement correct du logiciel SOUP. [Classes B, C]

NOTE Ces informations comprennent, par exemple, le type et la vitesse du processeur, le type et la taille de la mémoire, le type de SYSTÈME logiciel, les exigences relatives au logiciel de communication et d'affichage.

5.2.4 Re-EVALUATE MEDICAL DEVICE RISK ANALYSIS

The MANUFACTURER shall re-EVALUATE the MEDICAL DEVICE RISK ANALYSIS when software requirements are established and update it as appropriate. [Class A, B, C]

5.2.5 Update SYSTEM requirements

The MANUFACTURER shall ensure that existing requirements, including SYSTEM requirements, are re-EVALUATED and updated as appropriate as a result of the software requirements analysis ACTIVITY. [Class A, B, C]

5.2.6 Verify software requirements

The MANUFACTURER shall verify and document that the software requirements:

a) implement SYSTEM requirements including those relating to RISK CONTROL;

b) do not contradict one another;

c) are expressed in terms that avoid ambiguity;

d) are stated in terms that permit establishment of test criteria and performance of tests to determine whether the test criteria have been met;

e) can be uniquely identified; and

f) are traceable to SYSTEM requirements or other source.

[Class A, B, C]

NOTE This standard does not require the use of a formal specification language.

5.3 * Software ARCHITECTURAL design

5.3.1 Transform software requirements into an ARCHITECTURE

The MANUFACTURER shall transform the requirements for the MEDICAL DEVICE SOFTWARE into a documented ARCHITECTURE that describes the software's structure and identifies the SOFTWARE ITEMS. [Class B, C]

5.3.2 Develop an ARCHITECTURE for the interfaces of SOFTWARE ITEMS

The MANUFACTURER shall develop and document an ARCHITECTURE for the interfaces between the SOFTWARE ITEMS and the components external to the SOFTWARE ITEMS (both software and hardware), and between the SOFTWARE ITEMS. [Class B, C]

5.3.3 Specify functional and performance requirements of SOUP item

If a SOFTWARE ITEM is identified as SOUP, the MANUFACTURER shall specify functional and performance requirements for the SOUP item that are necessary for its intended use. [Class B, C]

5.3.4 Specify SYSTEM hardware and software required by SOUP item

If a SOFTWARE ITEM is identified as SOUP, the MANUFACTURER shall specify the SYSTEM hardware and software necessary to support the proper operation of the SOUP item. [Class B, C]

NOTE Examples include processor type and speed, memory type and size, SYSTEM software type, communication and display software requirements.

5.3.5 Identification des séparations nécessaires à la MAÎTRISE DU RISQUE

Le FABRICANT doit identifier les séparations entre ÉLÉMENTS LOGICIELS qui sont essentiels pour la MAÎTRISE DU RISQUE et indiquer la méthode permettant de s'assurer que la séparation est efficace. [Classe C]

NOTE Un exemple de séparation est l'exécution des ÉLÉMENTS LOGICIELS sur différents processeurs. L'efficacité de la séparation peut être assurée en évitant tout partage de ressources entre les processeurs.

5.3.6 Vérification de l'ARCHITECTURE du logiciel

Le FABRICANT doit vérifier et consigner que:

a) l'ARCHITECTURE du logiciel permet une mise en œuvre deS EXIGENCES DU SYSTÈME et des logiciels, y compris celles liées à la MAÎTRISE DU RISQUE;

b) l'ARCHITECTURE du logiciel est capable de prendre en charge les interfaces entre ÉLÉMENTS LOGICIELS ainsi qu'entre ÉLÉMENTS LOGICIELS et matériels; et

c) l'ARCHITECTURE du DISPOSITIF MÉDICAL assure le fonctionnement correct des éventuels ÉLÉMENTS LOGICIELS SOUP utilisés.

[Classes B, C]

5.4 * Conception détaillée du logiciel

5.4.1 Décomposition de l'ARCHITECTURE des LOGICIELS en UNITÉS LOGICIELLES

Le FABRICANT doit affiner l'ARCHITECTURE logicielle jusqu'à ce qu'elle soit représentée par les UNITÉS LOGICIELLES. [Classes B, C]

5.4.2 Elaboration de la conception détaillée de chaque UNITÉ LOGICIELLE

Le FABRICANT doit élaborer et documenter une conception détaillée de chaque UNITÉ LOGICIELLE de l'ÉLÉMENT LOGICIEL. [Classe C]

5.4.3 Elaboration de la conception détaillée pour les interfaces

Le FABRICANT doit élaborer et documenter une conception détaillée des interfaces éventuelles entre l'UNITÉ LOGICIELLE et les composants externes (matériels et logiciels), ainsi que pour les interfaces entre UNITÉS LOGICIELLES. [Classe C]

5.4.4 Vérification de la conception détaillée

Le FABRICANT doit vérifier et consigner que la conception détaillée du logiciel:

a) met en œuvre l'ARCHITECTURE du logiciel; et

b) est exempte de contradiction avec l'ARCHITECTURE du logiciel.

[Classe C]

5.5 * Mise en œuvre et vérification des UNITÉS LOGICIELLES

5.5.1 Mise en œuvre de chaque UNITÉ LOGICIELLE

Le FABRICANT doit mettre en œuvre chaque UNITÉ LOGICIELLE. [Classes A, B, C]

5.5.2 Etablissement du PROCESSUS DE VÉRIFICATION DES UNITÉS LOGICIELLES

Le FABRICANT doit établir des stratégies, méthodes et procédures pour la vérification de chaque UNITÉ LOGICIELLE. Lorsque la VÉRIFICATION est effectuée sur la base d'essais, les procédures d'essai doivent être ÉVALUÉES pour correction. [Classes B, C]

5.3.5 Identify segregation necessary for RISK CONTROL

The MANUFACTURER shall identify the segregation between SOFTWARE ITEMS that is essential to RISK CONTROL, and state how to ensure that the 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.

5.3.6 Verify software ARCHITECTURE

The MANUFACTURER shall verify and document that:

a) the ARCHITECTURE of the software implements SYSTEM and software requirements including those relating to RISK CONTROL;

b) the software ARCHITECTURE is able to support interfaces between SOFTWARE ITEMS and between SOFTWARE ITEMS and hardware; and

c) the MEDICAL DEVICE ARCHITECTURE supports proper operation of any SOUP items.

[Class B, C]

5.4 * Software detailed design

5.4.1 Refine SOFTWARE ARCHITECTURE into SOFTWARE UNITS

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

5.4.2 Develop detailed design for each SOFTWARE UNIT

The MANUFACTURER shall develop and document a detailed design for each SOFTWARE UNIT of the SOFTWARE ITEM. [Class C]

5.4.3 Develop detailed design for interfaces

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

5.4.4 Verify detailed design

The MANUFACTURER shall verify and document that the software detailed design:

a) implements the software ARCHITECTURE; and

b) is free from contradiction with the software ARCHITECTURE.

[Class C]

5.5 * SOFTWARE UNIT implementation and verification

5.5.1 Implement each SOFTWARE UNIT

The MANUFACTURER shall implement each SOFTWARE UNIT. [Class A, B, C]

5.5.2 Establish SOFTWARE UNIT VERIFICATION PROCESS

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

NOTE Il est admis de combiner en un seul plan et ensemble d'ACTIVITÉS, LES ESSAIS D'INTÉGRATION ET LES ESSAIS DU SYSTÈME LOGICIEL.

5.5.3 Critères d'acceptation de l'UNITÉ LOGICIELLE

Le FABRICANT doit établir des critères d'acceptation pour les UNITÉS LOGICIELLES avant intégration dans des ÉLÉMENTS LOGICIELS plus grands et s'assurer que les UNITÉS LOGICIELLES remplissent ces critères d'acceptation.

NOTE Exemples de critères d'acceptation:

– le code du logiciel met-il correctement en œuvre les exigences, y compris les mesures de MAÎTRISE DU RISQUE ?

– le code du logiciel est-il en contradiction avec les interfaces décrites dans les documents de conception détaillée de l'unité LOGICIELLE ?

– le code du logiciel est-il conforme aux procédures de programmation ou normes de codage établies ?

5.5.4 Critères supplémentaires d'acceptation de l'UNITÉ LOGICIELLE

Lorsqu'il participe à la conception, LE FABRICANT doit, selon le cas, introduire des critères supplémentaires d'acceptation suivants:

a) du séquencement approprié des événements;

b) du flux des données et du contrôle;

c) de l'affectation des ressources planifiées;

d) de la définition et détection des erreurs et reprise après erreur;

e) de l'initialisation des variables;

f) des autodiagnostics;

g) de la gestion de la mémoire et des dépassements de capacité de la mémoire; et

h) des conditions limites.

[Classe C]

5.5.5 VÉRIFICATION de l'UNITÉ LOGICIELLE

Le FABRICANT doit réaliser la VÉRIFICATION de l'UNITÉ LOGICIELLE et documenter les résultats.

[Classes B, C]

5.6 * Intégration et essai d'intégration du logiciel

5.6.1 Intégration des UNITÉS LOGICIELLES

Le FABRICANT doit intégrer les UNITÉS LOGICIELLES conformément au plan d'intégration (voir 5.1.5). [Classes B, C]

5.6.2 Vérification de l'intégration du logiciel

Le FABRICANT doit vérifier et enregistrer les aspects suivants de l'intégration du logiciel conformément au plan d'intégration (voir 5.1.5):

a) Les UNITÉS LOGICIELLES ont été intégrées dans les ÉLÉMENTS LOGICIELS dans le SYSTÈME LOGICIEL, et

b) Les éléments matériels, ÉLÉMENTS LOGICIELS et l'aide aux opérations manuelles (par exemple: interface homme machine, les menus d'aide en ligne, la reconnaissance vocale, les commandes vocales) du SYSTÈME correspondant ont bien été intégrés au SYSTÈME.

[Classes B, C]

NOTE Cette VÉRIFICATION concerne uniquement l'intégration des éléments, conformément au plan, et non leur performance prévue. Cette VÉRIFICATION est en général réalisée sous forme d'inspection.

NOTE It is acceptable to combine integration testing and SOFTWARE SYSTEM testing into a single plan and set of ACTIVITIES.

5.5.3 SOFTWARE UNIT acceptance criteria

The MANUFACTURER shall establish acceptance criteria for SOFTWARE UNITS prior to integration into larger SOFTWARE ITEMS as appropriate, and ensure that SOFTWARE UNITS meet acceptance criteria. [Class B, C]

NOTE Examples of acceptance criteria are:

– does the software code implement requirements including RISK CONTROL measures?

– is the software code free from contradiction with the interfaces documented in the detailed design of the SOFTWARE UNIT?

– does the software code conform to programming procedures or coding standards?

5.5.4 Additional SOFTWARE UNIT acceptance criteria

When present in the design, the MANUFACTURER shall include additional acceptance criteria as appropriate for:

a) proper event sequence;

b) data and control flow;

c) planned resource allocation;

d) fault handling (error definition, isolation, and recovery);

e) initialisation of variables;

f) self-diagnostics;

g) memory management and memory overflows; and

h) boundary conditions.

[Class C]

5.5.5 SOFTWARE UNIT VERIFICATION

The MANUFACTURER shall perform the SOFTWARE UNIT VERIFICATION and document the results. [Class B, C]

5.6 * Software integration and integration testing

5.6.1 Integrate SOFTWARE UNITS

The MANUFACTURER shall integrate the SOFTWARE UNITS in accordance with the integration plan (see 5.1.5). [Class B, C]

5.6.2 Verify software integration

The MANUFACTURER shall verify and record the following aspects of the software integration in accordance with the integration plan (see 5.1.5):

a) the SOFTWARE UNITS have been integrated into SOFTWARE ITEMS and the SOFTWARE SYSTEM; and

b) the hardware items, SOFTWARE ITEMS, and support for manual operations (e.g., human-equipment interface, on-line help menus, speech recognition, voice control) of the SYSTEM have been integrated into the SYSTEM.

[Class B, C]

NOTE This VERIFICATION is only that the items have been integrated according to the plan, not that they perform as intended. This VERIFICATION is most likely implemented by some form of inspection.

5.6.3 Essai du logiciel intégré

Le FABRICANT doit soumettre à des essais les ÉLÉMENTS LOGICIELS intégrés conformément au plan d'intégration (voir 5.1.5) et documenter les résultats correspondants. [Classes B, C]

5.6.4 Teneur des essais d'intégration

Pour les essais d'intégration du logiciel, le FABRICANT doit s'assurer que l'ÉLÉMENT LOGICIEL intégré s'exécute comme prévu.

[Classes B, C]

NOTE 1 Sont à considérer, par exemple:

- la fonctionnalité requise du logiciel;
- la mise en œuvre des mesures de MAÎTRISE DU RISQUE;
- la synchronisation et autre comportement spécifié;
- le fonctionnement spécifié des interfaces internes et externes; et
- les essais dans des conditions anormales incluant le mauvais usage prévisible.

NOTE 2 Il est admis de combiner en un seul plan et ensemble d'ACTIVITÉS, les essais d'intégration et les essais du SYSTÈME LOGICIEL.

5.6.5 Vérification des procédures d'essais d'intégration

Le FABRICANT doit ÉVALUER les procédures d'essais d'intégration pour s'assurer de leur bien fondé. [Classes B, C]

5.6.6 Réalisation d'ESSAIS DE RÉGRESSION

Lorsque des ÉLÉMENTS LOGICIELS sont intégrés, le FABRICANT doit réaliser un ESSAI DE RÉGRESSION approprié pour démontrer que des défauts n'ont pas été introduits dans le logiciel précédemment intégré. [Classes B, C]

5.6.7 Contenu de l'enregistrement des essais d'intégration

Le FABRICANT doit:

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

b) conserver suffisamment d'enregistrements pour permettre la reproduction de l'essai; et

c) identifier le contrôleur chargé d'effectuer l'essai.

[Classes B, C]

NOTE L'exigence b) pourrait être mise en œuvre en conservant par exemple:

- les spécifications d'essais montrant les actions requises et les résultats attendus,
- des enregistrements de l'équipement,
- des enregistrements de l'environnement d'essai (y compris les outils logiciels).

5.6.8 Utilisation du PROCESSUS de résolution des problèmes de logiciel

Le FABRICANT doit intégrer les ANOMALIES décelées lors de l'intégration et des essais d'intégration du logiciel dans un PROCESSUS de résolution des problèmes de logiciel [Classes B, C]

NOTE Voir Article 9.

5.6.3 Test integrated software

The MANUFACTURER shall test the integrated SOFTWARE ITEMS in accordance with the integration plan (see 5.1.5) and document the results. [Class B, C]

5.6.4 Integration testing content

For software integration testing, the MANUFACTURER shall address whether the integrated SOFTWARE ITEM performs as intended.

[Class B, C]

NOTE 1 Examples to be considered are:

- the required functionality of the software;
- implementation of RISK CONTROL measures;
- specified timing and other behaviour;
- specified functioning of internal and external interfaces; and
- testing under abnormal conditions including foreseeable misuse.

NOTE 2 It is acceptable to combine integration testing and SOFTWARE SYSTEM testing into a single plan and set of ACTIVITIES.

5.6.5 Verify integration test procedures

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

5.6.6 Conduct regression tests

When software items are integrated, the MANUFACTURER shall conduct REGRESSION TESTING appropriate to demonstrate that defects have not been introduced into previously integrated software. [Class B, C]

5.6.7 Integration test record contents

The MANUFACTURER shall:

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

b) retain sufficient records to permit the test to be repeated; and

c) identify the tester.

[Class B, C]

NOTE Requirement b) could be implemented by retaining, for example:

- test case specifications showing required actions and expected results;
- records of the equipment;
- records of the test environment (including software tools) used for test.

5.6.8 Use software problem resolution PROCESS

The MANUFACTURER shall enter ANOMALIES found during software integration and integration testing into a software problem resolution PROCESS. [Class B, C]

NOTE See Clause 9.

5.7 * Essais du SYSTÈME LOGICIEL

5.7.1 Etablissement d'essais pour les exigences du logiciel

Le FABRICANT doit établir et réaliser un ensemble d'essais exprimé en stimuli d'entrée, résultats attendus, critères de réussite/échec et en procédures d'exécution des essais du SYSTÈME LOGICIEL, de telle sorte que toutes les exigences du logiciel soient couvertes. [Classes B, C]

NOTE 1 Il est admis de combiner en un seul plan et ensemble d'ACTIVITÉS, les ESSAIS D'INTÉGRATION et les essais du SYSTÈME LOGICIEL. Il est également admis de soumettre les exigences logicielles à des essais au cours de phases antérieures.

NOTE 2 Il peut être réalisé, non seulement des essais séparés pour chaque exigence, mais aussi des essais de combinaison d'exigences, en particulier s'il existe des dépendances entre les exigences.

5.7.2 Utilisation du PROCESSUS de résolution des problèmes de logiciel

Le FABRICANT doit intégrer dans un PROCESSUS de résolution des problèmes de logiciel les ANOMALIES décelées au cours de l'essai du SYSTÈME LOGICIEL. [Classes B, C]

5.7.3 Contre-essais après modifications

Lorsque des modifications sont effectuées pendant les essais du SYSTÈME LOGICIEL, le FABRICANT doit:

a) recommencer les essais, effectuer des essais modifiés ou des essais supplémentaires, selon le cas, afin de vérifier l'efficacité de la modification pour la correction du problème;

b) effectuer des essais appropriés afin de démontrer que des effets secondaires non prévus n'ont pas été introduits; et

c) réaliser les ACTIVITÉS pertinentes de GESTION DES RISQUES comme définies en 7.4.

[Classes B, C]

5.7.4 Vérification des essais du SYSTÈME LOGICIEL

Le FABRICANT doit vérifier que:

a) les stratégies de VÉRIFICATION utilisées sont appropriées et les procédures d'essai sont appropriées;

b) les procédures d'essai du SYSTÈME LOGICIEL sont tracées vis-à-vis des exigences du logiciel;

c) toutes les exigences du logiciel ont été soumises à des essais ou VÉRIFIÉES par ailleurs; et

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

[Classes B, C]

5.7.5 Teneur des enregistrements d'essai du SYSTÈME LOGICIEL

Le FABRICANT doit:

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

b) conserver suffisamment d'enregistrements pour permettre la reproduction de l'essai; et

c) identifier le contrôleur chargé d'effectuer l'essai.

[Classes B, C]

NOTE L'exigence b) pourrait être mise en œuvre en conservant par exemple:

– les spécifications du jeu d'essais montrant les actions requises et les résultats attendus;

– des enregistrements du matériel d'essai; et

– des enregistrements de l'environnement d'essai (y compris les outils logiciels).

5.7 * SOFTWARE SYSTEM testing

5.7.1 Establish tests for software requirements

The MANUFACTURER shall establish and perform a set of tests, expressed as input stimuli, expected outcomes, pass/fail criteria and procedures, for conducting SOFTWARE SYSTEM testing, such that all software requirements are covered. [Class B, C]

NOTE 1 It is acceptable to combine integration testing and SOFTWARE SYSTEM testing into a single plan and set of ACTIVITIES. It is also acceptable to test software requirements in earlier phases.

NOTE 2 Not only separate tests for each requirement, but also tests of combinations of requirements can be performed, especially if dependencies between requirements exist.

5.7.2 Use software problem resolution PROCESS

The MANUFACTURER shall enter ANOMALIES found during software system testing into a software problem resolution PROCESS. [Class B, C]

5.7.3 Retest after changes

When changes are made during SOFTWARE SYSTEM testing, the MANUFACTURER shall:

a) repeat tests, perform modified tests or perform additional tests, as appropriate, to verify the effectiveness of the change in correcting the problem;

b) conduct testing appropriate to demonstrate that unintended side effects have not been introduced; and

c) perform relevant RISK MANAGEMENT ACTIVITIES as defined in 7.4.

[Class B, C]

5.7.4 Verify SOFTWARE SYSTEM testing

The MANUFACTURER shall verify that:

a) the VERIFICATION strategies and the test procedures used are appropriate;

b) SOFTWARE SYSTEM test procedures trace to software requirements;

c) all software requirements have been tested or otherwise VERIFIED; and

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

[Class B, C]

5.7.5 SOFTWARE SYSTEM test record contents

The MANUFACTURER shall:

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

b) retain sufficient records to permit the test to be repeated; and

c) identify the tester.

[Class B, C]

NOTE Requirement b) could be implemented by retaining, for example:

– test case specifications showing required actions and expected results;

– records of the equipment; and

– records of the test environment (including software tools) used for test.

5.8 * Diffusion du logiciel

5.8.1 Assurance de l'achèvement de la VÉRIFICATION du logiciel

Le FABRICANT doit s'assurer, avant diffusion du logiciel, que la VÉRIFICATION du logiciel est terminée et que les résultats ont été évalués. [Classes B, C]

5.8.2 Consignation des ANOMALIES résiduelles connues

Le FABRICANT doit documenter toutes les ANOMALIES résiduelles connues. [Classes B, C]

5.8.3 Évaluation des ANOMALIES résiduelles connues

Le FABRICANT doit s'assurer que toutes les ANOMALIES résiduelles connues ont été évaluées pour s'assurer qu'elles ne contribuent pas à un RISQUE inacceptable. [Classes B, C]

5.8.4 Consignation des VERSIONS diffusées

Le FABRICANT doit documenter la VERSION du PRODUIT LOGICIEL qui est diffusée. [Classes A, B, C]

5.8.5 Consignation de la manière dont le logiciel diffusé a été créé

Le FABRICANT doit documenter la procédure et l'environnement utilisés pour créer le logiciel diffusé. [Classes B, C]

5.8.6 Assurance de l'achèvement complet des ACTIVITÉS et des TÂCHES

Le FABRICANT doit s'assurer que toutes les ACTIVITÉS et TÂCHES sont complètement achevées et que la documentation associée est complète. [Classes B, C]

5.8.7 Archivage du logiciel

Le FABRICANT doit archiver:

a) le PRODUIT LOGICIEL et les ÉLÉMENTS DE CONFIGURATION; et

b) la documentation

pendant au moins une période déterminée comme étant la plus longue entre la durée de vie du dispositif telle que définie par le FABRICANT et un laps de temps spécifié par les exigences réglementaires pertinentes. [Classes B, C]

5.8.8 Assurance de la reproductibilité du logiciel diffusé

Le FABRICANT doit établir des procédures pour s'assurer que le PRODUIT LOGICIEL diffusé est bien livré de manière fiable au point d'utilisation, sans corruption ni changement non-autorisé. Ces procédures doivent concerner la production et le maniement des supports contenant le PRODUIT LOGICIEL, incluant selon le cas:

- la réplication,
- l'étiquetage,
- l'emballage,
- la protection,
- le stockage, et
- la livraison.

[Classes B, C]

5.8 * Software release

5.8.1 Ensure software VERIFICATION is complete

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

5.8.2 Document known residual ANOMALIES

The MANUFACTURER shall document all known residual ANOMALIES. [Class B, C]

5.8.3 EVALUATE known residual ANOMALIES

The MANUFACTURER shall ensure that all known residual ANOMALIES have been EVALUATED to ensure that they do not contribute to an unacceptable RISK. [Class B, C]

5.8.4 Document released VERSIONS

The MANUFACTURER shall document the VERSION of the SOFTWARE PRODUCT that is being released. [Class A, B, C]

5.8.5 Document how released software was created

The MANUFACTURER shall document the procedure and environment used to create the released software. [Class B, C]

5.8.6 Ensure activities and tasks are complete

The MANUFACTURER shall ensure that all ACTIVITIES and TASKS are complete along with all the associated documentation. [Class B, C]

5.8.7 Archive software

The MANUFACTURER shall archive:

a) the SOFTWARE PRODUCT and CONFIGURATION ITEMS; and

b) the documentation

for at least a period of time determined as the longer of: the life time of the device as defined by the MANUFACTURER or a time specified by relevant regulatory requirements. [Class B, C]

5.8.8 Assure repeatability of software release

The MANUFACTURER shall establish procedures to ensure that the released SOFTWARE PRODUCT can be reliably delivered to the point of use without corruption or unauthorised change. These procedures shall address the production and handling of media containing the SOFTWARE PRODUCT including as appropriate:

– replication,
– media labelling,
– packaging,
– protection,
– storage, and
– delivery.

[Class B, C]

6 PROCESSUS de maintenance du logiciel

6.1 * Etablissement du plan de maintenance du logiciel

Le FABRICANT doit établir un(des) plan(s) de maintenance du logiciel afin d'entreprendre les ACTIVITÉS et TÂCHES du PROCESSUS de maintenance. Le plan doit traiter des éléments suivants:

a) des procédures de:
   - réception,
   - documentation,
   - évaluation,
   - résolution et
   - suivi

   des retours d'information survenant après la diffusion du logiciel de DISPOSITIF MÉDICAL;

b) les critères permettant de déterminer si ces retours d'information constituent effectivement des problèmes ;

c) l'utilisation du PROCESSUS de GESTION DES RISQUES du logiciel;

d) l'utilisation du PROCESSUS de résolution des problèmes du logiciel afin d'analyser et résoudre les problèmes après diffusion du LOGICIEL DE DISPOSITIF MÉDICAL;

e) l'utilisation du PROCESSUS de gestion de la configuration du logiciel (Article 8) pour la gestion des modifications au SYSTÈME existant; et

f) les procédures d'évaluation et de mise en œuvre:
   - des mises à niveau,
   - des corrections de bogue,
   - des patches (rustines) et
   - de l'obsolescence

   des logiciels SOUP.

[Classes A, B, C]

6.2 * Analyse des problèmes et des modifications

6.2.1 Consignation et ÉVALUATION des retours d'information

6.2.1.1 Contrôle des retours d'information

Le FABRICANT doit contrôler tout retour d'information sur le PRODUIT LOGICIEL diffusé tant à l'intérieur de sa propre organisation que de la part des utilisateurs. [Classes A, B, C]

6.2.1.2 Consignation et ÉVALUATION des retours d'information

Les retours d'information doivent être documentés et ÉVALUÉS pour déterminer s'il y a un problème dans un PRODUIT LOGICIEL diffusé. Tout problème de ce type doit être enregistré comme RAPPORT DE PROBLÈME (voir Article 9) Les RAPPORTS DE PROBLÈME doivent comprendre les événements préjudiciables réels ou potentiels, ainsi que les écarts par rapport aux spécifications. [Classes A, B, C]

6.2.1.3 Évaluation des influences des RAPPORTS DE PROBLÈME sur la SÉCURITÉ

Chaque RAPPORT DE PROBLÈME doit être évalué afin de déterminer la manière dont il affecte la SÉCURITÉ d'un PRODUIT LOGICIEL diffusé et si une modification du PRODUIT LOGICIEL diffusé est nécessaire pour traiter le problème. [Classes A, B, C]

6 Software maintenance PROCESS

6.1 * Establish software maintenance plan

The MANUFACTURER shall establish a software maintenance plan (or plans) for conducting the ACTIVITIES and TASKS of the maintenance PROCESS. The plan shall address the following:

a) procedures for:
   - receiving,
   - documenting,
   - evaluating,
   - resolving and
   - tracking

   feedback arising after release of the MEDICAL DEVICE SOFTWARE;
b) criteria for determining whether feedback is considered to be a problem;
c) use of the software RISK MANAGEMENT PROCESS;
d) use of the software problem resolution PROCESS for analysing and resolving problems arising after release of the MEDICAL DEVICE SOFTWARE;
e) use of the software configuration management PROCESS (Clause 8) for managing modifications to the existing SYSTEM; and
f) procedures to EVALUATE and implement:
   - upgrades,
   - bug fixes,
   - patches and
   - obsolescence

   of SOUP.

[Class A, B, C]

6.2 * Problem and modification analysis

6.2.1 Document and EVALUATE feedback

6.2.1.1 Monitor feedback

The MANUFACTURER shall monitor feedback on released SOFTWARE PRODUCT from both inside its own organization and from users. [Class A, B, C]

6.2.1.2 Document and EVALUATE feedback

Feedback shall be documented and EVALUATED to determine whether a problem exists in a released SOFTWARE PRODUCT. Any such problem shall be recorded as a PROBLEM REPORT (see Clause 9). PROBLEM REPORTS shall include actual or potential adverse events, and deviations from specifications. [Class A, B, C]

6.2.1.3 Evaluate PROBLEM REPORT'S affects on SAFETY

Each PROBLEM REPORT shall be EVALUATED to determine how it affects the SAFETY of a released SOFTWARE PRODUCT and whether a change to the released SOFTWARE PRODUCT is needed to address the problem. [Class A, B, C]

6.2.2 Utilisation du PROCESSUS de résolution des problèmes du logiciel

Le FABRICANT doit utiliser le PROCESSUS de résolution des problèmes du logiciel (voir l'Article 9) pour traiter les RAPPORTS DE PROBLÈME. [Classes A, B, C]

NOTE Une fois cette ACTIVITÉ terminée, il convient de connaître toute modification de classe de sécurité dans le SYSTÈME LOGICIEL ou ses ÉLÉMENTS LOGICIELS.

6.2.3 Analyse des DEMANDES DE MODIFICATION

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

6.2.4 Approbation des DEMANDES DE MODIFICATION

Le FABRICANT doit ÉVALUER et approuver les DEMANDES DE MODIFICATION qui modifient les PRODUITS LOGICIELS diffusés. [Classes A, B, C]

6.2.5 Communication aux utilisateurs et aux organismes de réglementation

Le FABRICANT doit identifier les DEMANDES DE MODIFICATION approuvées qui affectent les PRODUITS LOGICIELS diffusés.

Si la réglementation locale l'exige, le FABRICANT doit informer les utilisateurs et les organismes de réglementation:

a) de tout problème affectant les PRODUITS LOGICIELS diffusés et les conséquences de la poursuite de leur utilisation sans modification; et

b) de la nature de toutes les modifications disponibles pour les produits LOGICIELS diffusés ainsi que la manière d'obtenir et d'installer ces modifications.

[Classes A, B, C]

6.3 * Mise en œuvre de la modification

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

Le FABRICANT doit utiliser le PROCESSUS de développement du logiciel (voir l'Article 5) ou un PROCESSUS de maintenance établi pour mettre en œuvre les modifications. [Classes A, B, C]

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

6.3.2 Rediffusion du SYSTÈME LOGICIEL modifié

Le FABRICANT doit diffuser les SYSTÈMES LOGICIELS modifiés conformément à 5.8. Les modifications peuvent être diffusées dans le cadre d'une rediffusion complète d'un SYSTÈME LOGICIEL ou comme un kit de modifications comprenant les ÉLÉMENTS LOGICIELS modifiés et les outils nécessaires pour installer les modifications pour un SYSTÈME LOGICIEL existant. [Classes A, B, C]

6.2.2 Use software problem resolution PROCESS

The MANUFACTURER shall use the software problem resolution PROCESS (see Clause 9) to address PROBLEM REPORTS. [Class A, B, C]

NOTE When this ACTIVITY has been done, any change of safety class in the SOFTWARE SYSTEM or its SOFTWARE ITEMS should be known.

6.2.3 Analyse CHANGE REQUESTS

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

6.2.4 CHANGE REQUEST approval

The MANUFACTURER shall EVALUATE and approve CHANGE REQUESTS which modify released SOFTWARE PRODUCTS. [Class A, B, C]

6.2.5 Communicate to users and regulators

The MANUFACTURER shall identify the approved CHANGE REQUESTS that affect released SOFTWARE PRODUCTS.

As required by local regulation, the MANUFACTURER shall inform users and regulators about:

a) any problem in released SOFTWARE PRODUCTS and the consequences of continued unchanged use; and

b) the nature of any available changes to released SOFTWARE PRODUCTS and how to obtain and install the changes.

[Class A, B, C]

6.3 * Modification implementation

6.3.1 Use established PROCESS to implement modification

The MANUFACTURER shall use the software development PROCESS (see Clause 5) or an established maintenance PROCESS to implement the modifications. [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

The MANUFACTURER shall release modified SOFTWARE SYSTEMS according to 5.8. Modifications may 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. [Class A, B, C]

7 * PROCESSUS DE GESTION DES RISQUES du logiciel

7.1 * Analyse du logiciel en termes de contribution à des situations dangereuses

7.1.1 Identification des ÉLÉMENTS LOGICIELS qui pourraient contribuer à une situation dangereuse

Le FABRICANT doit identifier les ÉLÉMENTS LOGICIELS qui pourraient contribuer à une situation dangereuse identifiée par l'ACTIVITÉ d'ANALYSE DU RISQUE du DISPOSITIF MÉDICAL comme défini dans l'ISO 14971 (voir 4.2). [Classes B, C]

NOTE La situation dangereuse pourrait être le résultat direct d'une défaillance de logiciel ou le résultat de la défaillance d'une mesure de MAÎTRISE DU RISQUE mise en œuvre dans le logiciel.

7.1.2 Identification des causes potentielles de contribution à une situation dangereuse

Le FABRICANT doit identifier les causes potentielles de la contribution de l'ÉLÉMENT LOGICIEL identifié ci-dessus à une situation dangereuse.

Le FABRICANT doit tenir compte des causes potentielles, incluant le cas échéant:

a) la spécification incorrecte ou incomplète de la fonctionnalité;

b) les défauts logiciels dans la fonctionnalité de l'ÉLÉMENT LOGICIEL identifié;

c) la défaillance ou résultat inattendu du logiciel SOUP;

d) les défaillances matérielles ou autres défauts logiciels qui pourraient donner lieu à un fonctionnement imprévisible du logiciel; et

e) un mauvais usage raisonnablement prévisible.

[Classes B, C]

7.1.3 ÉVALUATION des listes publiées D'ANOMALIES SOUP

Si une défaillance ou des résultats inattendus d'un logiciel SOUP est une cause potentielle de la contribution d'un ÉLÉMENT LOGICIEL à une situation dangereuse, le FABRICANT doit au minimum ÉVALUER toute liste d'ANOMALIES publiée par le fournisseur de l'élément du logiciel SOUP concernant la version de l'élément du logiciel SOUP utilisée dans le DISPOSITIF MÉDICAL, afin de déterminer si l'une des ANOMALIES connues entraîne une séquence d'événements qui pourrait donner lieu à une situation dangereuse. [Classes B, C]

7.1.4 Consignation des causes potentielles

Le FABRICANT doit documenter dans le DOSSIER DE GESTION DES RISQUES les causes potentielles de la contribution de l'élément LOGICIEL à une situation dangereuse (voir l'ISO 14971). [Classes B, C]

7.1.5 Consignation des séquences d'événements

Le FABRICANT doit documenter, dans le dossier de GESTION DES RISQUES, les séquences d'événements qui pourraient entraîner une situation dangereuse, telles qu'identifiées au 7.1.2. [Classes B, C]

7 * Software RISK MANAGEMENT PROCESS

7.1 * Analysis of software contributing to hazardous situations

7.1.1 Identify SOFTWARE ITEMS that could contribute to a hazardous situation

The MANUFACTURER shall identify SOFTWARE ITEMS that could contribute to a hazardous situation identified in the MEDICAL DEVICE RISK ANALYSIS ACTIVITY of ISO 14971 (see 4.2). [Class B, C]

NOTE The hazardous situation could be the direct result of software failure or the result of the failure of a RISK CONTROL measure that is implemented in software.

7.1.2 Identify potential causes of contribution to a hazardous situation

The MANUFACTURER shall identify potential causes of the SOFTWARE ITEM identified above contributing to a hazardous situation.

The MANUFACTURER shall consider potential causes including, as appropriate:

a) incorrect or incomplete specification of functionality;

b) software defects in the identified SOFTWARE ITEM functionality;

c) failure or unexpected results from SOUP;

d) hardware failures or other software defects that could result in unpredictable software operation; and

e) reasonably foreseeable misuse.

[Class B, C]

7.1.3 EVALUATE published SOUP ANOMALY lists

If failure or unexpected results from SOUP is a potential cause of the SOFTWARE ITEM contributing to a hazardous situation, the MANUFACTURER shall EVALUATE as a minimum any ANOMALY list published by the supplier of the SOUP item relevant to the VERSION of the SOUP item used in the MEDICAL DEVICE to determine if any of the known ANOMALIES result in a sequence of events that could result in a hazardous situation. [Class B, C]

7.1.4 Document potential causes

The MANUFACTURER shall document in the RISK MANAGEMENT FILE potential causes of the SOFTWARE ITEM contributing to a hazardous situation (see ISO 14971). [Class B, C]

7.1.5 Document sequences of events

The MANUFACTURER shall document in the RISK MANAGEMENT FILE sequences of events that could result in a hazardous situation that are identified in 7.1.2. [Class B, C]

7.2 Mesures DE MAÎTRISE DU RISQUE

7.2.1 Définition des mesures de MAÎTRISE DU RISQUE

Le FABRICANT doit définir et documenter des mesures de MAÎTRISE DU RISQUE pour chaque cause potentielle de l'ÉLÉMENT LOGICIEL contribuant à une situation dangereuse documentée dans le DOSSIER DE GESTION DES RISQUES. [Classe B, C]

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

7.2.2 Mesures de MAÎTRISE DU RISQUE mises en œuvre dans le logiciel

Si une mesure de MAÎTRISE DU RISQUE est mise en œuvre comme faisant partie des fonctions d'un ÉLÉMENT LOGICIEL, le FABRICANT doit:

a) inclure la mesure de MAÎTRISE DU RISQUE dans les exigences de logiciel;

b) attribuer une classe de SÉCURITÉ du logiciel à l'élément LOGICIEL sur la base des effets possibles du PHÉNOMÈNE DANGEREUX que contrôle la MAÎTRISE DU RISQUE; et

c) développer l'ÉLÉMENT LOGICIEL conformément à l'Article 5.

[Classes B, C]

NOTE Cette exigence fournit de plus amples détails sur les exigences de la MAÎTRISE DU RISQUE de l'ISO 14971.

7.3 VÉRIFICATION des mesures de MAÎTRISE DU RISQUE

7.3.1 Vérification des mesures de MAÎTRISE DU RISQUE

La mise en œuvre de chacune des mesures de MAÎTRISE DU RISQUE indiquées en 7.2 doit être VÉRIFIÉE, et cette VÉRIFICATION doit être documentée. [Classes B, C]

7.3.2 Consignation de toutes nouvelles séquences d'événements

Si une mesure de MAÎTRISE DU RISQUE est mise en œuvre comme ÉLÉMENT LOGICIEL, le FABRICANT doit ÉVALUER la mesure de MAÎTRISE DU RISQUE afin d'identifier et de documenter, dans le DOSSIER DE GESTION DES RISQUES, les éventuelles nouvelles séquences d'événements qui pourraient entraîner une situation dangereuse. [Classes B, C]

7.3.3 Consignation de la TRAÇABILITÉ

Le FABRICANT doit documenter la TRAÇABILITÉ des DANGERS liés au logiciel selon le cas:

a) de la situation dangereuse à l'ÉLÉMENT LOGICIEL;

b) de l'ÉLÉMENT LOGICIEL à la cause logicielle spécifique;

c) de la cause logicielle à la mesure de MAÎTRISE DU RISQUE; et

d) de la mesure dE MAÎTRISE DU RISQUE à la VÉRIFICATION de la mesure de MAÎTRISE DU RISQUE.

[Classes B, C]

NOTE Voir l'ISO 14971 – Rapport DE GESTION DES RISQUES.

7.2 RISK CONTROL measures

7.2.1 Define RISK CONTROL measures

For each potential cause of the software item contributing to a hazardous situation documented in the risk management file, the manufacturer shall define and document risk control measures. [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

If a RISK CONTROL measure is implemented as part of the functions of a SOFTWARE ITEM, the MANUFACTURER shall:

a) include the RISK CONTROL measure in the software requirements;

b) assign a software safety class to the SOFTWARE ITEM based on the possible effects of the HAZARD that the RISK CONTROL measure is controlling; and

c) develop the SOFTWARE ITEM in accordance with Clause 5.

[Class B, C]

NOTE This requirement provides additional detail for RISK CONTROL requirements of ISO 14971

7.3 VERIFICATION of RISK CONTROL measures

7.3.1 Verify RISK CONTROL measures

The implementation of each RISK CONTROL measure documented in 7.2 shall be VERIFIED, and this VERIFICATION shall be documented. [Class B, C]

7.3.2 Document any new sequences of events

If a RISK CONTROL measure is implemented as a SOFTWARE ITEM, the MANUFACTURER shall EVALUATE the RISK CONTROL measure to identify and document in the RISK MANAGEMENT FILE any new sequences of events that could result in a hazardous situation. [Class B, C]

7.3.3 Document TRACEABILITY

The MANUFACTURER shall document TRACEABILITY of software HAZARDS as appropriate:

a) from the hazardous situation to the SOFTWARE ITEM;

b) from the SOFTWARE ITEM to the specific software cause;

c) from the software cause to the RISK CONTROL measure; and

d) from the RISK CONTROL measure to the VERIFICATION of the RISK CONTROL measure.

[Class B, C]

NOTE See ISO 14971 – RISK MANAGEMENT report.

7.4 GESTION DES RISQUES des modifications du logiciel

7.4.1 Analyse des modifications apportées au LOGICIEL DE DISPOSITIF MÉDICAL en termes de SÉCURITÉ

Le FABRICANT doit analyser les modifications apportées au LOGICIEL DE DISPOSITIF MÉDICAL (y compris les logiciels SOUP) afin de déterminer si:

a) des causes potentielles supplémentaires contribuant à une situation dangereuse sont introduites; et

b) des mesures supplémentaires de MAÎTRISE DU RISQUE du logiciel sont nécessaires.

[Classes A, B, C]

7.4.2 Analyse de l'impact des modifications apportées au logiciel sur les mesures existantes de MAÎTRISE DU RISQUE

Le FABRICANT doit analyser les modifications apportées au logiciel, y compris celles apportées au logiciel SOUP, afin de déterminer si la modification du logiciel pourrait interférer avec des mesures existantes de MAÎTRISE DU RISQUE. [Classes B, C]

7.4.3 Réalisation des ACTIVITÉS DE GESTION DES RISQUES sur la base des analyses

Le FABRICANT doit réaliser les ACTIVITÉs pertinentes de GESTION DES RISQUES définies en 7.1, 7.2 et 7.3 sur la base de ces analyses. [Classes B, C]

8 * PROCESSUS de gestion de configuration du logiciel

8.1 * Identification de la configuration

8.1.1 Etablissement des moyens d'identification des ÉLÉMENTS DE CONFIGURATION

Le FABRICANT doit établir un plan pour l'identification univoque des ÉLÉMENTS DE CONFIGURATION et leurs VERSIONS à maîtriser dans le cadre du projet. Ce plan doit inclure les autres PRODUITS LOGICIELS ou entités tels que les logiciels SOUP et la documentation. [Classes A, B, C]

8.1.2 Identification des logiciels SOUP

Pour chaque ÉLÉMENT DE CONFIGURATION de logiciel SOUP utilisé, y compris les bibliothèques standards, le FABRICANT doit documenter:

a) le titre,

b) le FABRICANT, et

c) la désignation unique du logiciel SOUP

de chaque ÉLÉMENT DE CONFIGURATION de logiciel SOUP à utiliser. [Classes A, B, C]

NOTE La désignation unique du logiciel SOUP pourrait être par exemple, une VERSION, une date de diffusion, un numéro de correctif ou une désignation de mise à niveau.

8.1.3 Identification de la documentation de configuration du SYSTÈME

Le FABRICANT doit documenter l'ensemble des éléments de CONFIGURATION et leurs VERSIONS qui constituent la configuration du SYSTÈME LOGICIEL. [Classes A, B, C]

7.4 RISK MANAGEMENT of software changes

7.4.1 Analyse changes to MEDICAL DEVICE SOFTWARE with respect to SAFETY

The MANUFACTURER shall analyse changes to the MEDICAL DEVICE SOFTWARE (including SOUP) to determine whether:

a) additional potential causes are introduced contributing to a hazardous situation; and

b) additional software RISK CONTROL measures are required.

[Class A, B, C]

7.4.2 Analyse impact of software changes on existing RISK CONTROL measures

The MANUFACTURER shall analyse changes to the software, including changes to SOUP, to determine whether the software modification could interfere with existing RISK CONTROL measures. [Class B, C]

7.4.3 Perform RISK MANAGEMENT ACTIVITIES based on analyses

The MANUFACTURER shall perform relevant RISK MANAGEMENT ACTIVITIES defined in 7.1, 7.2 and 7.3 based on these analyses. [Class B, C]

8 * Software configuration management PROCESS

8.1 * Configuration identification

8.1.1 Establish means to identify CONFIGURATION ITEMS

The MANUFACTURER shall establish a scheme for the unique identification of CONFIGURATION ITEMS and their VERSIONS to be controlled for the project. This scheme shall include other SOFTWARE PRODUCTS or entities such as SOUP and documentation. [Class A, B, C]

8.1.2 Identify SOUP

For each SOUP CONFIGURATION ITEM being used, including standard libraries, the MANUFACTURER shall document:

a) the title,

b) the MANUFACTURER, and

c) the unique SOUP designator

of each SOUP CONFIGURATION ITEM being used. [Class A, B, C]

NOTE The unique SOUP designator could be, for example, a VERSION, a release date, a patch number or an upgrade designation.

8.1.3 Identify SYSTEM configuration documentation

The MANUFACTURER shall document the set of CONFIGURATION ITEMS and their VERSIONS that comprise the SOFTWARE SYSTEM configuration. [Class A, B, C]

8.2 * Maîtrise des modifications

8.2.1 Approbation des DEMANDES DE MODIFICATION

Le FABRICANT ne doit modifier des ÉLÉMENTS DE CONFIGURATION qu'en réponse à une DEMANDE DE MODIFICATION approuvée. [Classes A, B, C]

NOTE 1 La décision d'approuver une DEMANDE DE MODIFICATION peut être intégrée au PROCESSUS de maîtrise des modifications ou faire partie d'un autre PROCESSUS. Le présent paragraphe exige uniquement qu'une modification soit approuvée avant sa mise en œuvre.

NOTE 2 Différents PROCESSUS d'acceptation peuvent être utilisés pour les DEMANDES DE MODIFICATION à différentes étapes du cycle de vie, comme indiqué dans les plans, voir 5.1.1 e) et 6.1 e).

8.2.2 Mise en œuvre des modifications

Le FABRICANT doit mettre en œuvre la modification comme spécifié dans la DEMANDE DE MODIFICATION. Le FABRICANT doit identifier et réaliser toute ACTIVITÉ qu'il est nécessaire de répéter du fait de la modification, y compris les changements de la classification de SÉCURITÉ DU LOGICIEL des SYSTÈMES LOGICIELS et des ÉLÉMENTS LOGICIELS. [Classe A, B, C]

NOTE Le présent paragraphe précise la manière dont il convient de mettre en œuvre la modification pour assurer une maîtrise appropriée des modifications. Il n'implique en aucune manière que la mise en œuvre fait partie intégrante du PROCESSUS de maîtrise des modifications. Il est recommandé que la mise en œuvre utilise des PROCESSUS planifiés, voir 5.1.1 e) et 6.1 e).

8.2.3 Vérification des modifications

Le FABRICANT doit vérifier la modification, y compris la répétition de toute VÉRIFICATION invalidée par la modification et la prise en compte du 5.7.3 et du 9.7. [Classes A, B, C]

NOTE Ce paragraphe exige uniquement que les modifications soient vérifiées. Il n'implique en aucune manière que la VÉRIFICATION fasse partie intégrante du PROCESSUS de maîtrise des modifications. Il est recommandé que la VÉRIFICATION utilise des PROCESSUS planifiés, voir 5.1.1 e) et 6.1 e).

8.2.4 Prévision des moyens de TRAÇABILITÉ de la modification

Le FABRICANT doit créer un enregistrement d'audit permettant à chaque:

a) DEMANDE DE MODIFICATION;

b) RAPPORT DE PROBLÈME pertinent; et

c) approbation de la DEMANDE DE MODIFICATION

d'être tracée. [Classes A, B, C]

8.3 * Documentation relative à l'état de la configuration

Le FABRICANT doit conserver des enregistrements récupérables de l'historique des ÉLÉMENTS DE CONFIGURATION maîtrisés y compris la configuration du SYSTÈME. [Classes A, B, C]

9 * PROCESSUS de résolution de problème logiciel

9.1 Elaboration des RAPPORTS DE PROBLÈME

Le FABRICANT doit rédiger un RAPPORT DE PROBLÈME pour chaque problème détecté dans un PRODUIT LOGICIEL. Les RAPPORTS DE PROBLÈME doivent être classés comme suit:

a) le type;

EXEMPLE 1 correctif, préventif ou adaptation à un nouvel environnement

8.2 * Change control

8.2.1 Approve CHANGE REQUESTS

The MANUFACTURER shall change CONFIGURATION ITEMS only in response to an approved CHANGE REQUEST. [Class A, B, C]

NOTE 1 The decision to approve a CHANGE REQUEST can be integral to the change control PROCESS or part of another PROCESS. This subclause only requires that approval of a change precede its implementation.

NOTE 2 Different acceptance PROCESSES can be used for CHANGE REQUESTS at different stages of the life cycle, as stated in plans, see 5.1.1 e) and 6.1 e).

8.2.2 Implement changes

The MANUFACTURER shall implement the change as specified in the CHANGE REQUEST. The MANUFACTURER shall identify and perform any ACTIVITY that needs to be repeated as a result of the change, including changes to the software safety classification of SOFTWARE SYSTEMS and SOFTWARE ITEMS. [Class A, B, C]

NOTE This subclause states how the change should be implemented to achieve adequate change control. It does not imply that the implementation is an integral part of the change control PROCESS. Implementation should use planned PROCESSES, see 5.1.1 e) and 6.1 e).

8.2.3 Verify changes

The MANUFACTURER shall verify the change, including repeating any VERIFICATION that has been invalidated by the change and taking into account 5.7.3 and 9.7. [Class A, B, C]

NOTE This subclause only requires that changes be VERIFIED. It does not imply that VERIFICATION is an integral part of the change control PROCESS. VERIFICATION should use planned PROCESSES, see 5.1.1 e) and 6.1 e).

8.2.4 Provide means for TRACEABILITY of change

The MANUFACTURER shall create an audit trail whereby each:

a) CHANGE REQUEST;

b) relevant PROBLEM REPORT; and

c) approval of the CHANGE REQUEST

can be traced. [Class A, B, C]

8.3 * Configuration status accounting

The MANUFACTURER shall retain retrievable records of the history of controlled CONFIGURATION ITEMS including SYSTEM configuration. [Class A, B, C]

9 * Software problem resolution PROCESS

9.1 Prepare PROBLEM REPORTS

The MANUFACTURER shall prepare a PROBLEM REPORT for each problem detected in a SOFTWARE PRODUCT. PROBLEM REPORTS shall be classified as follows:

a) type;

EXAMPLE 1 corrective, preventive, or adaptive to new environment

b) le domaine d'application; et

EXEMPLE 2 étendue de la modification, nombre de modèles de dispositifs concernés, accessoires pris en charge concernés, ressources impliquées, temps nécessaire pour la modification

c) la criticité.

EXEMPLE 3 effet sur les performances, LA SÉCURITÉ ou LA SÛRETÉ

[Classes A, B, C]

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

9.2 Etude du problème

Le FABRICANT doit:

a) étudier le problème et si possible en identifier les causes;

b) ÉVALUER la pertinence du problème en termes de SÉCURITÉ en utilisant le PROCESSUS de GESTION DES RISQUEs du logiciel (Article 7);

c) documenter le résultat de la recherche et de l'évaluation; et

d) déclencher la ou les DEMANDES DE MODIFICATION nécessaires pour corriger le problème, ou justifier le fait de n'entreprendre aucune action.

[Classes A, B, C]

NOTE Il n'est pas nécessaire qu'un problème ait été corrigé pour que le FABRICANT soit conforme au PROCESSUS de résolution des problèmes du logiciel, si le problème ne concerne pas la SÉCURITÉ.

9.3 Information des parties concernées

Le FABRICANT doit informer les parties concernées de l'existence du problème, selon le cas. [Classes A, B, C]

NOTE Les problèmes peuvent être découverts avant ou après la diffusion, à l'intérieur de l'organisation du FABRICANT ou à l'extérieur. Le FABRICANT détermine les parties concernées en fonction de la situation.

9.4 Utilisation du processus de la maîtrise des modifications

Le FABRICANT doit approuver et mettre en œuvre toutes les DEMANDES DE MODIFICATION, en observant les exigences du processus de la maîtrise des modifications (voir 8.2). [Classes A, B, C]

9.5 Conservation des enregistrements

Le FABRICANT doit conserver des enregistrements des RAPPORTS DE PROBLÈMES et de leur résolution y compris leur VÉRIFICATION.

Le FABRICANT doit mettre à jour le dossier de GESTION DES RISQUES selon le cas (voir 7.4). [Classes A, B, C]

9.6 Analyse de tendance pour les problèmes

Le FABRICANT doit effectuer une analyse permettant de détecter les tendances dans les RAPPORTS DE PROBLÈMES. [Classes A, B, C]

b) scope; and

EXAMPLE 2 size of change, number of device models affected, supported accessories affected, resources involved, time to change

c) criticality.

EXAMPLE 3 effect on performance, SAFETY, or SECURITY

[Class A, B, C]

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

9.2 Investigate the problem

The MANUFACTURER shall:

a) investigate the problem and if possible identify the causes;

b) EVALUATE the problem's relevance to SAFETY using the software RISK MANAGEMENT PROCESS (Clause 7);

c) document the outcome of the investigation and evaluation; and

d) create a CHANGE REQUEST(S) for actions needed to correct the problem, or document the rationale for taking no action.

[Class A, B, C]

NOTE A problem does not have to be corrected for the MANUFACTURER to comply with the software problem resolution PROCESS, provided that the problem is not relevant to SAFETY.

9.3 Advise relevant parties

The MANUFACTURER shall advise relevant parties of the existence of the problem, as appropriate.

[Class A, B, C]

NOTE Problems can be discovered before or after release, inside the MANUFACTURER'S organisation or outside it. The MANUFACTURER determines the relevant parties depending on the situation.

9.4 Use change control process

The MANUFACTURER shall approve and implement all CHANGE REQUESTS, observing the requirements of the change control PROCESS (see 8.2). [Class A, B, C]

9.5 Maintain records

The MANUFACTURER shall maintain records of PROBLEM REPORTS and their resolution including their VERIFICATION.

The MANUFACTURER shall update the RISK MANAGEMENT FILE as appropriate (see 7.4) [Class A, B, C]

9.6 Analyse problems for trends

The MANUFACTURER shall perform analysis to detect trends in PROBLEM REPORTS. [Class A, B, C]

9.7 VÉRIFICATION de la résolution des problèmes du logiciel

Le FABRICANT doit vérifier les résolutions de problèmes afin de s'assurer:

a) que le problème a été résolu et que le RAPPORT DE PROBLÈMES a été clos;

b) que les tendances préjudiciables ont été inversées;

c) que les DEMANDES DE MODIFICATION ont été mises en œuvre dans les PRODUITS LOGICIELS et ACTIVITÉS concernées; et

d) que des problèmes supplémentaires n'ont pas été introduits.

[Classes A, B, C]

9.8 Teneur de la documentation d'essai

Lors d'essais, de contre-essais ou d'essais de régression des ÉLÉMENTS LOGICIELS et SYSTÈMES, suite à une modification, le FABRICANT doit inclure dans la documentation d'essai:

a) les résultats d'essai;

b) les ANOMALIES décelées;

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

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

e) les outils d'essai pertinents;

f) la date de l'essai; et

g) l'identification du contrôleur ayant effectué l'essai.

[Classes A, B, C]

9.7 Verify software problem resolution

The MANUFACTURER shall verify resolutions to determine whether:

a) problem has been resolved and the PROBLEM REPORT has been closed;

b) adverse trends have been reversed;

c) CHANGE REQUESTS have been implemented in the appropriate SOFTWARE PRODUCTS and ACTIVITIES; and

d) additional problems have been introduced.

[Class A, B, C]

9.8 Test documentation contents

When testing, retesting or REGRESSION TESTING SOFTWARE ITEMS and SYSTEMS following a change, the MANUFACTURER shall include in the test documentation:

a) test results;

b) ANOMALIES found;

c) the VERSION of software tested;

d) relevant hardware and software test configurations;

e) relevant test tools;

f) date tested; and

g) identification of the tester.

[Class A, B, C]

Annexe A
(informative)

Justification des exigences de la présente norme

La présente annexe fournit une justification des articles de la présente norme.

A.1 Justification du raisonnement

La principale exigence de la présente norme est qu'un ensemble de PROCESSUS doit être suivi pour le développement et la maintenance des LOGICIELS DE DISPOSITIFS MÉDICAUX et que le choix des PROCESSUS soit adapté aux RISQUES encourus par le patient et autre personne concernée. Ceci procède de la conviction selon laquelle les essais de logiciel ne sont pas suffisants pour déterminer que son fonctionnement est sûr.

Les PROCESSUS exigés par la présente norme s'inscrivent dans deux catégories:

– Les PROCESSUS qui sont exigés pour déterminer les RISQUES résultant du fonctionnement de chaque ÉLÉMENT LOGICIEL dans le logiciel;

– Les PROCESSUS qui sont exigés pour atteindre un taux de probabilité de défaut de logiciel suffisamment bas pour chaque ÉLÉMENT LOGICIEL, choisi sur la base de la détermination desdits RISQUEs.

La présente norme exige que la première catégorie soit appliquée à tout LOGICIEL DE DISPOSITIF MÉDICAL et la seconde catégorie porte sur des ÉLÉMENTS LOGICIELS choisis.

Ainsi, il convient que toute revendication de conformité à la présente norme inclue une analyse des RISQUES écrite qui identifie les séquences prévisibles d'événements impliquant le logiciel et qui peuvent donner lieu à une situation dangereuse (voir l'ISO 14971). Il convient par conséquent d'inclure dans cette ANALYSE DE RISQUE les DANGERS qui pourraient être directement induits par le logiciel (par exemple, la fourniture d'informations propres à induire en erreur et qui pourraient donner lieu à l'administration d'un traitement inadéquat).

Toutes les ACTIVITÉS qui sont exigées comme faisant partie de la première catégorie des PROCESSUS sont identifiées dans le texte normatif comme « [Classes A, B, C] », indiquant ainsi qu'elles sont exigées quelle que soit la classification du logiciel auquel elles s'appliquent.

LES ACTIVITÉS sont exigées pour toutes les classes A, B et C pour les raisons suivantes:

– l'ACTIVITÉ génère un plan qui s'applique à la GESTION DES RISQUES ou à la classification de SÉCURITÉ du logiciel;

– l'ACTIVITÉ génère un élément de sortie qui est un élément d'entrée pour la GESTION DES RISQUES ou la classification de SÉCURITÉ du logiciel;

– l'ACTIVITÉ fait partie de la GESTION DES RISQUES ou de la classification de SÉCURITÉ du logiciel;

– l'ACTIVITÉ établit un système d'administration, de documentation ou de tenue des enregistrements qui vient à l'appui de la GESTION DES RISQUES ou de la classification de SÉCURITÉ du logiciel;

– l'ACTIVITÉ est en général entreprise à un moment où la classification du logiciel concernée n'est pas connue;

– l'ACTIVITÉ peut donner lieu à une modification qui pourrait invalider la classification actuelle de SÉCURITÉ du logiciel concernée. Ceci comprend la découverte et l'analyse de problèmes liés à la SÉCURITÉ après diffusion du logiciel.

Annex A
(informative)

Rationale for the requirements of this standard

Rationale for the clauses of this standard is provided in this annex.

A.1 Rationale

The primary requirement of this standard is that a set of PROCESSES be followed in the development and maintenance of MEDICAL DEVICE SOFTWARE, and that the choice of PROCESSES be appropriate to the RISKS to the patient and other people. This follows from the belief that testing of software is not sufficient to determine that it is safe in operation.

The PROCESSES required by this standard fall into two categories:

– PROCESSES which are required to determine the RISKS arising from the operation of each SOFTWARE ITEM in the software;

– PROCESSES which are required to achieve an appropriately low probability of software failure for each SOFTWARE ITEM, chosen on the basis of these determined RISKS.

This standard requires the first category to be performed for all MEDICAL DEVICE SOFTWARE and the second category to be performed for selected SOFTWARE ITEMS.

A claim of compliance with this standard should therefore include a documented RISK ANALYSIS that identifies foreseeable sequences of events that include software and that can result in a hazardous situation (see ISO 14971). HAZARDS that can be indirectly caused by software (for example, by providing misleading information that could cause inappropriate treatment to be administered) should be included in this RISK ANALYSIS.

All ACTIVITIES that are required as part of the first category of PROCESSES are identified in the normative text as "[Class A, B, C]", indicating that they are required irrespective of the classification of the software to which they apply.

ACTIVITIES are required for all classes A, B, and C for the following reasons:

– the ACTIVITY produces a plan relevant to RISK MANAGEMENT or software safety classification;

– the ACTIVITY produces an output that is an input to RISK MANAGEMENT or software safety classification;

– the ACTIVITY is a part of RISK MANAGEMENT or software safety classification;

– the ACTIVITY establishes an administration system, documentation or record-keeping system that supports RISK MANAGEMENT or software safety classification;

– the ACTIVITY normally takes place when the classification of the related software is unknown;

– the ACTIVITY can cause a change that could invalidate the current software safety classification of the associated software. This includes the discovery and analysis of safety related problems after release.

D'autres PROCESSUS sont exigés seulement pour les SYSTÈMES LOGICIELS ou LES ÉLÉMENTS LOGICIELS classés dans les classes de SÉCURITÉ de logiciel B ou C. Les ACTIVITÉS exigées comme faisant partie de ce PROCESSUS sont identifiées dans le texte normatif comme « [Classe B, C] », ou « [Classe C] » indiquant ainsi qu'elles sont exigées de manière sélective en fonction de la classification du logiciel auquel elles s'appliquent.

Les ACTIVITÉS sont exigées sélectivement pour les logiciels de classe B ou C pour les raisons suivantes:

– l'ACTIVITÉ améliore la fiabilité du logiciel en exigeant plus de détails ou plus de rigueur de conception, d'essai ou autre vérification;
– l'ACTIVITÉ est une ACTIVITÉ administrative qui vient à l'appui d'une autre ACTIVITÉ exigée pour les classes B ou C;
– l'ACTIVITÉ prend en charge la correction de problèmes relatifs à la SÉCURITÉ;
– l'ACTIVITÉ produit des enregistrements de la conception, de la mise en œuvre, de la VÉRIFICATION et de la diffusion de logiciels relatifs à la SÉCURITÉ.

Les ACTIVITÉS sont exigées sélectivement pour les logiciels de classe C pour les raisons suivantes:

– L'ACTIVITÉ apporte une amélioration supplémentaire à la fiabilité du logiciel en exigeant plus de détails ou plus de rigueur ou plus d'attention à des points spécifiques de la conception, des essais ou autre VÉRIFICATION

Il est à noter que tous les PROCESSUS et ACTIVITÉS définis dans la présente norme sont considérés déterminants pour assurer le développement et la maintenance de logiciels de grande qualité. L'omission de nombre de ces PROCESSUS et ACTIVITÉS en tant qu'exigences pour les logiciels de classe A qui ne peuvent par définition donner lieu à un PHÉNOMÈNE DANGEREUX ne signifie pas que ces PROCESSUS et ACTIVITÉS ne seraient pas importants ou qu'ils ne sont pas recommandés. Leur omission permet de reconnaître que la SÉCURITÉ et l'efficacité de logiciels qui ne peuvent donner lieu à des DANGERS peuvent être facilement assurées, principalement par une ACTIVITÉ de validation globale lors de la conception du DISPOSITIF MÉDICAL (qui est hors du domaine d'application de la présente norme) ainsi que par de simples contrôles du cycle de vie du logiciel.

Other PROCESSES are required only for SOFTWARE SYSTEMS or SOFTWARE ITEMS classified in software safety classes B or C. ACTIVITIES required as parts of these PROCESSES are identified in the normative text as "[Class B, C]", or "[Class C]" indicating that they are required selectively depending on the classification of the software to which they apply.

ACTIVITIES are required selectively for software in classes B and C for the following reasons:

– the ACTIVITY enhances the reliability of the software by requiring more detail or more rigor in the design, testing or other VERIFICATION;
– the ACTIVITY is an administrative ACTIVITY that supports another ACTIVITY required for classes B or C;
– the ACTIVITY supports the correction of safety-related problems;
– the ACTIVITY produces records of design, implementation, VERIFICATION and release of safety-related software.

ACTIVITIES are required selectively for software in class C for the following reasons:

– the ACTIVITY further enhances the reliability of the software by requiring more detail, or more rigour, or attention to specific issues in the design, testing or other VERIFICATION

Note that all PROCESSES and ACTIVITIES defined in this standard are considered valuable in assuring the development and maintenance of high quality software. The omission of many of these PROCESSES and ACTIVITIES as requirements for software in class A that cannot by definition cause a HAZARD should not imply that these PROCESSES and ACTIVITIES would not be of value or are not recommended. Their omission is intended to recognize that software that cannot cause a HAZARD can be easily assured of SAFETY and effectiveness primarily through overall validation ACTIVITY during the design of a MEDICAL DEVICE (which is outside the scope of this standard) and through some simple software life cycle controls.

A.2 Récapitulatif des exigences par classe

Le Tableau A.1 résume les classes de sécurité de logiciel qui sont attribuées à chaque exigence. Ce tableau est informatif et fourni uniquement pour commodité. La section normative identifie les classes de sécurité de logiciel pour chaque exigence.

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

Articles et paragraphes		Classe A	Classe B	Classe C
Article 4	Toutes les exigences	X	X	X
5.1	5.1.1, 5.1.2, 5.1.3, 5.1.6, 5.1.7, 5.1.8, 5.1.9	X	X	X
	5.1.5, 5.1.10, 5.1.11		X	X
	5.1.4			X
5.2	5.2.1, 5.2.2, 5.2.4, 5.2.5, 5.2.6	X	X	X
	5.2.3		X	X
5.3	5.3.1, 5.3.2, 5.3.3, 5.3.4, 5.3.6		X	X
	5.3.5			X
5.4	5.4.1		X	X
	5.4.2, 5.4.3, 5.4.4			X
5.5	5.5.1	X	X	X
	5.5.2, 5.5.3, 5.5.5		X	X
	5.5.4			X
5.6	Toutes les exigences		X	X
5.7	Toutes les exigences		X	X
5.8	5.8.4	X	X	X
	5.8.1, 5.8.2, 5.8.3, 5.8.5, 5.8.6, 5.8.7, 5.8.8		X	X
6.1	6.1	X	X	X
6.2	6.2.1, 6.2.2, 6.2.4, 6.2.5	X	X	X
	6.2.3		X	X
6.3	Toutes les exigences	X	X	X
7.1	Toutes les exigences		X	X
7.2	Toutes les exigences		X	X
7.3	Toutes les exigences		X	X
7.4	7.4.1	X	X	X
	7.4.2, 7.4.3		X	X
Article 8	Toutes les exigences	X	X	X
Article 9	Toutes les exigences	X	X	X

A.2 Summary of requirements by class

Table A.1 summarizes which software safety classes are assigned to each requirement. This table is informative and only provided for convenience. The normative section identifies the software safety classes for each requirement.

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		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
5.8	5.8.4	X	X	X
	5.8.1, 5.8.2, 5.8.3, 5.8.5, 5.8.6, 5.8.7, 5.8.8		X	X
6.1	6.1	X	X	X
6.2	6.2.1, 6.2.2, 6.2.4, 6.2.5	X	X	X
	6.2.3		X	X
6.3	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

Annexe B
(informative)

Lignes directrices relatives aux dispositions de la présente norme

B.1 Domaine d'application

B.1.1 Objet

L'objet de la présente norme est de fournir un PROCESSUS de développement qui produira de manière homogène des LOGICIELS DE DISPOSITIFS MÉDICAUX de grande qualité et sûrs. Pour cela, la présente norme identifie les ACTIVITÉS et TÂCHES minimales qui peuvent être réalisées pour acquérir la certitude que le logiciel a été développé d'une manière qui produira vraisemblablement des PRODUITS LOGICIELS hautement fiables et sûrs.

La présente annexe fournit des lignes directrices pour l'application des exigences de la présente norme. Elle n'ajoute aucune information supplémentaire ni ne modifie les exigences de la présente norme. Cette annexe peut être utilisée pour mieux comprendre les exigences de la présente norme.

Il est à noter que dans la présente norme, les ACTIVITÉS sont définies dans des paragraphes référencés dans les PROCESSUS et que les TÂCHES sont définies dans les ACTIVITÉS correspondantes. Par exemple, les ACTIVITÉS définies pour le PROCESSUS de développement du logiciel sont la planification du développement du LOGICIEL, l'analyse des exigences du logiciel, la conception ARCHITECTURALE du logiciel, la conception détaillée du logiciel, le codage du logiciel, la mise en œuvre et la VÉRIFICATION de L'UNITÉ LOGICIELLE, l'intégration et les essais d'intégration du logiciel, les essais du SYSTÈME LOGICIEL et la diffusion du logiciel. Les TÂCHES dans ces ACTIVITÉs sont les exigences individuelles.

La présente norme n'exige pas un MODÈLE particulier de CYCLE DE VIE DE DÉVELOPPEMENT DU LOGICIEL. Cependant, la conformité à la présente norme implique effectivement des dépendances entre PROCESSUS, car les éléments d'entrée d'un PROCESSUS donné sont générés par un autre PROCESSUS. Il convient par exemple, de compléter la classification de la sécurité du logiciel du SYSTÈME LOGICIEL une fois que l'ANALYSE DU RISQUE a établi les dommages qui pourraient résulter de la défaillance du SYSTÈME LOGICIEL.

Du fait de ces dépendances logiques entre PROCESSUS, il est plus facile de décrire les PROCESSUS dans la présente norme comme une séquence, ce qui implique un modèle de cycle de vie de type «en cascade» ou «à passage unique». Cependant, d'autres cycles de vie peuvent également être utilisés. Certaines stratégies (modèles) de développement, qui sont définies dans l'ISO/CEI 12207 [9], comportent les stratégies suivantes (voir également le Tableau B.1):

– En cascade: La stratégie à «passage unique» également appelée «en cascade», consiste à réaliser le PROCESSUS de développement en une seule fois. De manière plus simple: on détermine les besoins du client, on définit les exigences, on conçoit le SYSTÈME, on met en œuvre le SYSTÈME, on le soumet aux essais, on le corrige et on le livre.
– Incrémentielle: La stratégie «incrémentielle» consiste à déterminer les besoins du client et à définir les exigences du SYSTÈME puis à entreprendre le reste du développement en une séquence d'éléments de construction. Le premier élément de construction intègre une partie des capacités prévues, l'élément de construction suivant ajoute des capacités supplémentaires et ainsi de suite jusqu'à ce que le SYSTÈME soit complet.
– Evolutive: La stratégie «évolutive» consiste également à développer un SYSTÈME en éléments de construction mais diffère de la stratégie incrémentielle du fait qu'elle considère de prime abord que le besoin de l'utilisateur n'est pas pleinement compris et que toutes les exigences ne peuvent être définies en amont. Dans cette stratégie, les besoins du client et les exigences du SYSTÈME sont partiellement définis en amont mais sont ensuite définis à chaque élément de construction suivant.

Annex B
(informative)

Guidance on the provisions of this standard

B.1 Scope

B.1.1 Purpose

The purpose of this standard is to provide a development PROCESS that will consistently produce high quality, safe MEDICAL DEVICE SOFTWARE. To accomplish this, the standard identifies the minimum ACTIVITIES and TASKS that need to be accomplished to provide confidence that the software has been developed in a manner that is likely to produce highly reliable and safe SOFTWARE PRODUCTS.

This annex provides guidance for the application of the requirements of this standard. It does not add to, or otherwise change, the requirements of this standard. This annex can be used to better understand the requirements of this standard.

Note that in this standard, ACTIVITIES are subclauses called out within the PROCESSES and TASKS are defined within the ACTIVITIES. For example, the ACTIVITIES defined for the software development PROCESS are software development planning, software requirements analysis, software ARCHITECTURAL design, software detailed design, SOFTWARE UNIT implementation and VERIFICATION, software integration and integration testing, SOFTWARE SYSTEM testing, and software release. The TASKS within these ACTIVITIES are the individual requirements.

This standard does not require a particular SOFTWARE DEVELOPMENT LIFE CYCLE MODEL. However, compliance with this standard does imply dependencies between PROCESSES, because inputs of a PROCESS are generated by another PROCESS. For example, the software safety classification of the SOFTWARE SYSTEM should be completed after the RISK ANALYSIS PROCESS has established what HARM could arise from failure of the SOFTWARE SYSTEM.

Because of such logical dependencies between processes, it is easiest to describe the processes in this standard in a sequence, implying a “waterfall” or “once-through” life cycle model. However, other life cycles can also be used. Some development (model) strategies as defined at ISO/IEC 12207 [9] include (see also Table B.1):

– Waterfall. The “once-through" strategy, also called “waterfall”, consists of performing the development PROCESS a single time. Simplistically: determine customer needs, define requirements, design the SYSTEM, implement the system, test, fix and deliver.
– Incremental: The “incremental” strategy determines customer needs and defines the SYSTEM requirements, then performs the rest of the development in a sequence of builds. The first build incorporates part of the planned capabilities, the next build adds more capabilities, and so on, until the SYSTEM is complete.
– Evolutionary: The “evolutionary” strategy also develops a SYSTEM in builds but differs from the incremental strategy in acknowledging that the user need is not fully understood and all requirements cannot be defined up front. In this strategy, customer needs and SYSTEM requirements are partially defined up front, then are refined in each succeeding build.

Tableau B.1 – Stratégies (modèle) de développement telles que définies dans l'ISO/CEI 12207

Stratégie de développement	Définition a priori de toutes les exigences?	Cycles de développement multiples ?	Diffusion d'un logiciel intermédiaire ?
En cascade (Passage unique)	Oui	Non	Non
Incrémentielle (Amélioration du produit pré-planifiée)	Oui	Oui	Probablement
Evolutive	Non	Oui	Oui

Quel que soit le cycle de vie choisi, il est nécessaire de maintenir les dépendances logiques entre éléments de sortie du PROCESSUS tels que les spécifications, les documents de conception et les logiciels. Le modèle de cycle de vie en cascade atteint cet objectif en retardant le début du PROCESSUS jusqu'à ce que les éléments d'entrée de ce processus soient terminés et approuvés.

D'autres cycles de vie, notamment les cycles de vie évolutifs, permettent la production des éléments de sortie du PROCESSUS avant disponibilité de tous les éléments d'entrée de ce PROCESSUS. Il est possible par exemple de spécifier un nouvel ÉLÉMENT LOGICIEL, de le classer, de le mettre en œuvre et de le vérifier avant que l'ensemble de l'ARCHITECTURE DU LOGICIEL n'ait été finalisé. Le RISQUE encouru par de tels cycles de vie est qu'une modification ou un développement apporté à un élément de sortie d'un PROCESSUS donné invalidera l'élément de sortie d'un autre PROCESSUS. Par conséquent, tous les cycles de vie utilisent un système exhaustif de gestion de la configuration pour s'assurer que tous les éléments de sortie d'un PROCESSUS sont amenés à un état homogène et que les dépendances sont bien maintenues.

Quel que soit le cycle de vie de développement du logiciel utilisé, les principes suivants sont d'une importance primordiale:

– il convient de maintenir en un état cohérent tous les éléments de sortie de PROCESSUS; lorsqu'un élément de sortie de PROCESSUS est créé ou modifié, il convient de mettre à jour rapidement tous les éléments de sortie de PROCESSUS correspondants afin d'assurer leur cohérence les uns vis-à-vis des autres et de maintenir explicitement ou implicitement les dépendances exigées par la présente norme;
– il convient que tous les éléments de sortie des PROCESSUS soient disponibles lorsque cela est nécessaire en tant qu'éléments d'entrée pour poursuivre les travaux sur le logiciel.
– avant diffusion d'un LOGICIEL DE DISPOSITIF MÉDICAL, il convient d'assurer la cohérence des éléments de sortie des PROCESSUS et d'observer toutes les dépendances entre éléments de sortie de PROCESSUS explicitement ou implicitement exigées par la présente norme.

B.1.2 Domaine d'application

La présente norme s'applique au développement et à la maintenance de LOGICIELS DE DISPOSITIFS MÉDICAUX ainsi qu'au développement et à la maintenance d'un DISPOSITIF MÉDICAL qui comprend un logiciel SOUP.

L'utilisation de la présente norme exige que le FABRICANT applique une GESTION DES RISQUES du DISPOSITIF MÉDICAL qui soit conforme à l'ISO 14971. Par conséquent, lorsque l'ARCHITECTURE du système de DISPOSITIF MÉDICAL comprend un composant acquis (il pourrait s'agir d'un composant acheté ou d'un composant de provenance inconnue), tel qu'une imprimante ou un traceur qui comprend un logiciel SOUP, ce composant acquis passe sous la responsabilité du FABRICANT et doit être inclus dans la GESTION DES RISQUES du DISPOSITIF MÉDICAL. On suppose que par la réalisation correcte de la GESTION DES RISQUES pour le DISPOSITIF MÉDICAL, le FABRICANT comprendrait le composant et reconnaîtrait qu'il comprend un logiciel SOUP. Le FABRICANT utilisant la présente norme ferait alors appel au PROCESSUS de GESTION DES RISQUES du logiciel dans le cadre du PROCESSUS DE GESTION DES RISQUES du DISPOSITIF MÉDICAL.

Table B.1 – Development (model) strategies as defined in ISO/IEC 12207

Development Strategy	Define all requirements first?	Multiple development cycles?	Distribute interim software?
Waterfall (Once-through)	yes	no	no
Incremental (Preplanned product improvement)	yes	yes	maybe
Evolutionary	no	yes	yes

Whichever life cycle is chosen it is necessary to maintain the logical dependencies between PROCESS outputs such as specifications, design documents and software. The waterfall life cycle model achieves this by delaying the start of a PROCESS until the inputs for that PROCESS are complete and approved.

Other life cycles, particularly evolutionary life cycles, permit PROCESS outputs to be produced before all the inputs for that PROCESS are available. For example, a new SOFTWARE ITEM can be specified, classified, implemented and VERIFIED before the whole software ARCHITECTURE has been finalised. Such life cycles carry the RISK that a change or development in one PROCESS output will invalidate another PROCESS output. All life cycles therefore use a comprehensive configuration management system to ensure that all PROCESS outputs are brought to a consistent state and the dependencies maintained.

The following principles are important regardless of the software development life cycle used:

– All PROCESS outputs should be maintained in a consistent state; whenever any PROCESS output is created or changed, all related PROCESS outputs should be updated promptly to maintain their consistency with each other and to maintain all dependencies explicitly or implicitly required by this standard;
– all PROCESS outputs should be available when needed as input to further work on the software.
– before any MEDICAL DEVICE SOFTWARE is released, all PROCESS outputs should be consistent with each other and all dependencies between PROCESS outputs explicitly or implicitly required by this standard should be observed.

B.1.2 Field of application

This standard applies to the development and maintenance of MEDICAL DEVICE SOFTWARE as well as the development and maintenance of a MEDICAL DEVICE that includes SOUP.

The use of this standard requires the MANUFACTURER to perform MEDICAL DEVICE RISK MANAGEMENT that is compliant with ISO 14971. Therefore, when the MEDICAL DEVICE SYSTEM ARCHITECTURE includes an acquired component (this could be a purchased component or a component of unknown provenance), such as a printer/plotter that includes SOUP, the acquired component becomes the responsibility of the MANUFACTURER and must be included in the RISK MANAGEMENT of the MEDICAL DEVICE. It is assumed that through proper performance of MEDICAL DEVICE RISK MANAGEMENT, the MANUFACTURER would understand the component and recognize that it includes SOUP. The MANUFACTURER using this standard would invoke the software RISK MANAGEMENT PROCESS as part of MEDICAL DEVICE RISK MANAGEMENT PROCESS.

La maintenance du LOGICIEL de DISPOSITIF MÉDICAL diffusé s'applique au retour d'expérience post-production acquise avec le LOGICIEL de DISPOSITIF MÉDICAL. La maintenance du logiciel inclut la combinaison de toutes les actions techniques et administratives, y compris les opérations de surveillance, pour agir sur un rapport de problème afin de maintenir ou de remettre un élément dans un état lui permettant d'accomplir une fonction requise, aussi bien que des DEMANDES DE MODIFICATION liées à des PRODUITS LOGICIELS diffusés. Par exemple, ceci inclut la rectification d'un problème, le rapport réglementaire, une nouvelle validation et l'action préventive. Voir ISO/CEI 14764 [10].

B.2 Références normatives

La norme ISO/CEI 90003 [11] fournit des lignes directrices pour l'application d'un système de management de la qualité au développement du logiciel. Ces lignes directrices ne sont pas exigées par la présente norme mais hautement recommandées.

B.3 Termes et définitions

Dans toute la mesure du possible, des termes ont été spécifiés sur la base de définitions extraites de normes internationales.

La présente norme a choisi d'utiliser trois termes pour décrire la décomposition d'un SYSTÈME LOGICIEL (niveau le plus élevé). Le SYSTÈME LOGICIEL peut être un sous-système du DISPOSITIF MÉDICAL (voir la CEI 60601-1-4 [2]) ou un DISPOSITIF MÉDICAL à part entière. Le niveau inférieur qui ne peut être à nouveau décomposé à des fins d'essai ou de gestion de la configuration de logiciel est l'UNITÉ LOGICIELLE. Tous les niveaux de décomposition, y compris les niveaux supérieur et inférieur, peuvent être appelés ÉLÉMENTS LOGICIELS. Un SYSTÈME LOGICIEL est ainsi constitué de un ou plusieurs ÉLÉMENTS LOGICIELS et chaque ÉLÉMENT LOGICIEL est constitué d'une ou de plusieurs UNITÉS LOGICIELLES ou d'ÉLÉMENTS LOGICIELS décomposables. Il incombe au FABRICANT de fournir la définition et la granularité des ÉLÉMENTS LOGICIELS et des UNITÉS LOGICIELLES. Si la définition de ces termes reste vague, ils pourront être appliqués aux nombreuses et diverses méthodes de développement et types de logiciels utilisés dans les DISPOSITIFS MÉDICAUX.

B.4 Exigences générales

Il n'existe pas de méthode connue permettant d'assurer une SÉCURITÉ à 100 % pour tout type de logiciel.

Trois principes majeurs permettent de promouvoir la SÉCURITÉ des LOGICIELS de DISPOSITIFS MÉDICAUX:

– la GESTION DES RISQUES;
– le management de la qualité;
– l'ingénierie logicielle.

Pour le développement et la maintenance de logiciels sûrs de dispositifs médicaux, il est nécessaire de mettre en place une GESTION DES RISQUES comme partie intégrante d'un système de management de la qualité et utilisée comme cadre global pour l'application de méthodes et techniques appropriées d'ingénierie logicielle. La combinaison de ces trois concepts permet à un FABRICANT de DISPOSITIF MÉDICAL de suivre un PROCESSUS décisionnel clairement structuré et reproductible de manière homogène pour promouvoir la SÉCURITÉ du LOGICIEL DE DISPOSITIF MÉDICAL.

The maintenance of released MEDICAL DEVICE SOFTWARE applies to the post-production experience with the MEDICAL DEVICE SOFTWARE. Software maintenance includes the combination of all technical and administrative means, including supervision actions, to act on problem reports to retain an item in, or restore it to, a state in which it can perform a required function as well as modification requests related to released SOFTWARE PRODUCT(S). For example, this includes problem rectification, regulatory reporting, re-validation and preventive action. See ISO/IEC 14764 [10].

B.2 Normative references

ISO/IEC 90003 [11] provides guidance for applying a quality management system to software development. This guidance is not required by this standard but is highly recommended.

B.3 Terms and definitions

Where possible, terms have been defined using definitions from international standards.

This standard chose to use three terms to describe the decomposition of a SOFTWARE SYSTEM (top level). The SOFTWARE SYSTEM can be a subsystem of the MEDICAL DEVICE (see IEC 60601-1-4 [2]) or a MEDICAL DEVICE in its own right. The lowest level that is not further decomposed for the purposes of testing or software configuration management 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 definition and granularity of the SOFTWARE ITEMS and SOFTWARE UNITS. Leaving these terms vague allows one to apply them to the many different development methods and types of software used in MEDICAL DEVICES.

B.4 General requirements

There is no known method to guarantee 100 % SAFETY for any kind of software.

There are three major principles which promote SAFETY for MEDICAL DEVICE SOFTWARE:

– RISK MANAGEMENT;
– quality management;
– software engineering.

For the development and maintenance of safe MEDICAL DEVICE SOFTWARE it is necessary to establish RISK MANAGEMENT as an integral part of a quality management system as an overall framework for the application of appropriate software engineering methods and techniques. The combination of these three concepts allows a MEDICAL DEVICE MANUFACTURER to follow a clearly structured and consistently repeatable decision-making PROCESS to promote SAFETY for MEDICAL DEVICE SOFTWARE.

B.4.1 Système de management de la qualité

Un ensemble discipliné et efficace de PROCESSUS logiciels comprend des PROCESSUS organisationnels tels que la gestion, l'infrastructure, l'amélioration et la formation. Ces PROCESSUS ont été omis de la présente norme pour éviter les duplications et pour se concentrer sur l'ingénierie logicielle. Ces PROCESSUS font l'objet d'un système de management de la qualité. L'ISO 13485 [7] est une Norme Internationale qui est spécifiquement dédiée à l'application des concepts de l'assurance qualité aux DISPOSITIFS MÉDICAUX. La conformité aux exigences du système de management de la qualité de l'ISO 13485 ne constitue pas automatiquement une conformité aux exigences réglementaires, nationales ou régionales. Il incombe au FABRICANT d'identifier et d'établir la conformité aux exigences réglementaires applicables.

B.4.2 GESTION DES RISQUES

Le développement des logiciels participe suffisamment aux ACTIVITÉS de GESTION DES RISQUES pour assurer que tous les RISQUES raisonnablement prévisibles, associés au LOGICIEL DE DISPOSITIF MÉDICAL sont effectivement pris en compte.

Plutôt que de tenter de définir un PROCESSUS de GESTION DES RISQUES approprié dans cette norme relative à l'ingénierie logicielle, il est exigé que le FABRICANT applique un PROCESSUS DE GESTION DES RISQUES qui soit conforme à l'ISO 14971, dans la mesure où cette dernière traite explicitement de la GESTION DES RISQUES pour les DISPOSITIFS MÉDICAUX. Les ACTIVITÉS de la GESTION DES RISQUES spécifiques au logiciel résultant de DANGERS dont le logiciel est une des causes sont identifiées dans un PROCESSUS explicatif décrit dans l'Article 7.

B.4.3 Classification de sécurité du logiciel

Le RISQUE associé au logiciel en tant que partie constituante d'un DISPOSITIF MÉDICAL, en tant qu'accessoire d'un dispositif MÉDICAL, ou en tant que DISPOSITIF MÉDICAL à part entière, est utilisé comme l'élément d'entrée d'un plan de classification de la sécurité logicielle qui permet alors de déterminer les PROCESSUS à utiliser au cours du développement et de la maintenance du logiciel.

Le RISQUE est défini comme étant une combinaison de la gravité du DOMMAGE et de la probabilité de sa survenance. Cependant, il n'existe pas de consensus quant à la manière de déterminer la probabilité des défaillances logicielles sur la base de méthodes statistiques conventionnelles. Par conséquent, dans la présente norme, la classification du SYSTÈME LOGICIEL est fondée sur la gravité du DANGER résultant de la défaillance du logiciel, en supposant que la défaillance aura lieu. Les SYSTÈMES LOGICIELS qui contribuent à la mise en œuvre des mesures de MAÎTRISE DU RISQUE sont classés en fonction de la gravité du DANGER correspondant.

Si un SYSTÈME LOGICIEL est décomposé en ÉLÉMENTS LOGICIELS, alors chaque ÉLÉMENT LOGICIEL peut avoir sa propre classification de sécurité du logiciel.

Le RISQUE lié à la défaillance d'un ÉLÉMENT LOGICIEL ne peut être déterminé que:

– si le rôle de l'ÉLÉMENT LOGICIEL, en termes de fonctionnalité et d'interfaces avec d'autres ÉLÉMENTS LOGICIELS et matériels, est défini par une ARCHITECTURE du système et une ARCHITECTURE du logiciel;

– si les modifications au SYSTÈME sont maîtrisées;

– après réalisation d'une ANALYSE DE RISQUE sur l'ARCHITECTURE et application des mesures de MAÎTRISE DU RISQUE spécifiées.

La présente norme exige que soit entrepris le nombre minimal d'ACTIVITÉS qui permettront de satisfaire les conditions ci-dessus pour toutes les classes de logiciel.

B.4.1 Quality management system

A disciplined and effective set of software PROCESSES includes organizational PROCESSES such as management, infrastructure, improvement, and training. To avoid duplication and to focus this standard on software engineering, these PROCESSES have been omitted from this standard. These PROCESSES are covered by a quality management system. ISO 13485 [7] is an International Standard that is specifically intended for applying the concepts of quality management to MEDICAL DEVICES. Conformance to ISO 13485 quality management system requirements does not automatically constitute conformity with national or regional regulatory requirements. It is the MANUFACTURER'S responsibility to identify and establish compliance with relevant regulatory requirements.

B.4.2 RISK MANAGEMENT

Software development participates in RISK MANAGEMENT ACTIVITIES sufficiently to ensure that all reasonably foreseeable RISKS associated with the MEDICAL DEVICE SOFTWARE are considered.

Rather than trying to define an appropriate RISK MANAGEMENT PROCESS in this software engineering standard, it is required that the MANUFACTURER apply a RISK MANAGEMENT PROCESS that is compliant with ISO 14971, which deals explicitly with RISK MANAGEMENT for MEDICAL DEVICES. Specific software RISK MANAGEMENT ACTIVITIES resulting from HAZARDS that have software as a contributing cause are identified in a supporting PROCESS described in Clause 7.

B.4.3 Software safety classification

The RISK associated with software as a part of a MEDICAL DEVICE, as an accessory to a MEDICAL DEVICE, or as a MEDICAL DEVICE in its own right, is used as the input to a software safety classification scheme, which then determines the PROCESSES to be used during the development and maintenance of software.

RISK is considered to be a combination of the severity of injury and the probability of its occurrence. However, there is no consensus on how to determine the probability of occurrence of software failures using traditional statistical methods. In this standard, therefore, SOFTWARE SYSTEM classification is based on the severity of the HAZARD resulting from failure of the software, assuming that the failure will occur. SOFTWARE SYSTEMS that contribute to the implementation of RISK CONTROL measures are classified based on the severity of the HAZARD they are controlling.

If a SOFTWARE SYSTEM is decomposed into SOFTWARE ITEMS, then each SOFTWARE ITEM can have its own software safety classification.

It is only possible to determine the RISK associated with failure of a SOFTWARE ITEM:

– if a SYSTEM ARCHITECTURE and a software ARCHITECTURE define the role of the SOFTWARE ITEM in terms of its purpose and its interfaces with other software and hardware items;
– if changes to the SYSTEM are controlled;
– after RISK ANALYSIS has been done on the ARCHITECTURE and RISK CONTROL measures specified.

This standard requires the minimum number of ACTIVITIES that will achieve the above conditions for all classes of software.

L'achèvement d'une ACTIVITÉ d'ARCHITECTURE DU LOGICIEL est le point le plus en amont du développement où l'ensemble complet des ÉLÉMENTS LOGICIELS est défini et que l'ACTIVITÉ de GESTION DES RISQUES a identifié la manière dont les ÉLÉMENTS LOGICIELS sont liés à la SÉCURITÉ. Il s'agit par conséquent du point le plus précoce au niveau duquel les ÉLÉMENTS LOGICIELS peuvent être définitivement classés en fonction de leur effet sur la SÉCURITÉ.

Ce point correspond au point pour lequel la MAÎTRISE DU RISQUE commence dans l'ISO 14971.

Avant d'atteindre ce point, le PROCESSUS de GESTION DES RISQUES identifie les mesures de MAÎTRISE DU RISQUE ARCHITECTURAL en ajoutant par exemple des sous-systèmes de protection ou en réduisant les possibilités de défaillance logicielles qui peuvent entraîner des DOMMAGES. Après ce point, le PROCESSUS DE GESTION DES RISQUES utilise des PROCESSUS destinés à réduire la probabilité de défaillance des ÉLÉMENTS LOGICIELS. En d'autres termes, la classification d'un ÉLÉMENT LOGICIEL prescrit l'application à cet élément de mesures de MAÎTRISE DU RISQUE basée sur les PROCESSUS.

Il est probable que les FABRICANTS considèrent qu'il est utile de classer les logiciels avant ce point pour, par exemple, s'attacher plus particulièrement à des domaines de recherche particuliers, mais il convient que cette classification soit considérée comme préliminaire et ne soit pas utilisée pour justifier l'omission des PROCESSUS.

Le plan de classification de SÉCURITÉ du logiciel ne signifie pas un alignement sur la classification des RISQUES de l'ISO 14971. Alors que le plan de l'ISO 14971 classe les RISQUES en fonction de leur gravité et de leur probabilité, le plan de classification de SÉCURITÉ du logiciel définit les SYSTÈMES LOGICIELS et ÉLÉMENTS LOGICIELS en fonction des PROCESSUS à appliquer au cours de leur développement et de leur maintenance.

Au fur et à mesure que la conception évolue, de nouveaux RISQUES pourraient apparaître. Il est par conséquent recommandé d'appliquer la GESTION DES RISQUES comme partie intégrante du PROCESSUS de développement. Ceci permet de développer une conception architecturale qui identifie un ensemble complet d'ÉLÉMENTS LOGICIELS, y compris ceux dont il est exigé une fonctionnalité correcte pour assurer un fonctionnement en toute SÉCURITÉ et ceux qui évitent que les défauts n'aboutissent à des DOMMAGES.

Il convient que l'ARCHITECTURE DU LOGICIEL favorise la différentiation des ÉLÉMENTS LOGICIELS qui sont requis pour un fonctionnement en toute SÉCURITÉ et qu'elle décrive les méthodes utilisées pour assurer une différentiation effective de ces ÉLÉMENTS LOGICIELS.

Comme indiqué en B.3, la présente norme a choisi d'utiliser trois termes pour décrire la décomposition d'un SYSTÈME LOGICIEL (le niveau le plus élevé).

La Figure B.1 illustre le découpage possible d'ÉLÉMENTS LOGICIELS dans un SYSTÈME LOGICIEL donné et la manière dont les classes de SÉCURITÉ du logiciel seraient appliquées au groupe d'ÉLÉMENTS LOGICIELS de la décomposition.

The end of the software ARCHITECTURE ACTIVITY is the earliest point in the development when the full set of SOFTWARE ITEMS is defined and the RISK MANAGEMENT ACTIVITY has identified how the SOFTWARE ITEMS relate to SAFETY. This is therefore the earliest point at which SOFTWARE ITEMS can be classified definitively according to their SAFETY role.

This point corresponds to the point where RISK CONTROL is begun in ISO 14971.

Before this point, the RISK MANAGEMENT PROCESS identifies ARCHITECTURAL RISK CONTROL measures, for example adding protective subsystems, or reducing the opportunities for software failures to cause HARM. After this point, the RISK MANAGEMENT PROCESS uses PROCESSES aimed at reducing the probability of failure of SOFTWARE ITEMS. In other words, the classification of a SOFTWARE ITEM specifies PROCESS-based RISK CONTROL measures to be applied to that item.

It is expected that MANUFACTURERS will find it useful to classify software before this point, for example to focus attention on areas to be investigated, but such classification should be regarded as preliminary and should not be used to justify the omission of PROCESSES.

The software safety classification scheme is not intended to align with the RISK classifications of ISO 14971. Whereas the ISO 14971 scheme classifies RISK according to their severity and likelihood, the software safety classification scheme classifies SOFTWARE SYSTEMS and SOFTWARE ITEMS according to the PROCESSES to be applied in their development and maintenance.

As the design evolves, new RISKS might become evident. Therefore, RISK MANAGEMENT should be applied as an integral part of the development PROCESS. This permits the development of an ARCHITECTURAL design that identifies a complete set of SOFTWARE ITEMS, including those that are required to function correctly to assure safe operation and those that prevent faults from causing HARM.

The software ARCHITECTURE should promote segregation of software items that are required for safe operation and should describe the methods used to ensure effective segregation of those SOFTWARE ITEMS.

As stated in B.3, this standard chooses to use three terms to describe the decomposition of a SOFTWARE SYSTEM (top level).

Figure B.1 illustrates the possible partitioning for SOFTWARE ITEMS within a SOFTWARE SYSTEM and how the software safety classes would be applied to the group of SOFTWARE ITEMS in the decomposition.

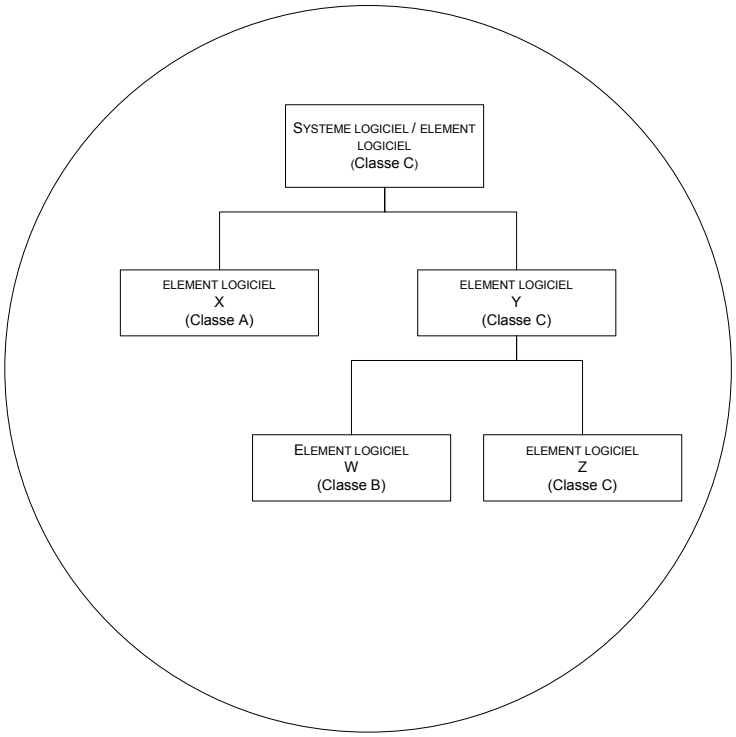

Figure B.1 – Exemple de découpage d'ÉLÉMENTS LOGICIELS

Dans cet exemple, le FABRICANT sait, du fait du type de LOGICIEL DE DISPOSITIF MÉDICAL à développer, que la classification préliminaire de SÉCURITÉ du logiciel pour le SYSTÈME LOGICIEL est la classe C de SÉCURITÉ du logiciel. Pendant la conception de l'ARCHITECTURE du logiciel, le FABRICANT a décidé de découper le SYSTÈME, comme illustré, en 3 ÉLÉMENTS LOGICIELS – X, W et Z. Le FABRICANT est capable d'isoler de l'ÉLÉMENT LOGICIEL Z toutes les contributions du SYSTÈME LOGICIEL aux DANGERS qui pourraient entraîner la mort ou une blessure grave, et de l'ÉLÉMENT LOGICIEL W toutes les contributions restantes du SYSTÈME LOGICIEL aux DANGERS qui pourraient entraîner une blessure légère. L'ÉLÉMENT LOGICIEL W est de classe B de SÉCURITÉ du logiciel et l'ÉLÉMENT LOGICIEL Z est de classe C de SÉCURITÉ du logiciel. Par conséquent, L'ÉLÉMENT LOGICIEL Y doit être de Classe C, conformément à 4.3 d). Le SYSTÈME LOGICIEL est également, du fait de cette exigence, de classe C de SÉCURITÉ du logiciel. L'ÉLÉMENT LOGICIEL X a été défini en classe A de SÉCURITÉ du logiciel. Le FABRICANT est capable de justifier la séparation entre les ÉLÉMENTS LOGICIELS X et Y, ainsi qu'entre les ÉLÉMENTS LOGICIELS W et Z, pour en assurer l'intégrité. Si la division n'est pas possible, les ÉLÉMENTS LOGICIELS X et Y doivent être classés dans la classe C de sécurité du logiciel.

B.5 PROCESSUS de développement du logiciel

B.5.1 Planification du développement du logiciel

L'objectif de cette ACTIVITÉ est de planifier les TÂCHES de développement du logiciel de manière à réduire les RISQUES causés par le logiciel, de communiquer les procédures et les objectifs aux membres de l'équipe de développement et de s'assurer que les exigences de qualité du système pour le LOGICIEL DE DISPOSITIF MÉDICAL sont remplies.

L'ACTIVITÉ de planification du développement du logiciel peut documenter des TÂCHES dans un plan unique ou dans plusieurs plans. Il est admis que certains FABRICANTS établissent des politiques et des procédures qui s'appliquent au développement de tout LOGICIEL DE DISPOSITIF MÉDICAL qui leur appartient. Dans ce cas, le plan peut simplement référencer les politiques et procédures existantes. Il est admis que certains FABRICANTS élaborent un plan ou un ensemble

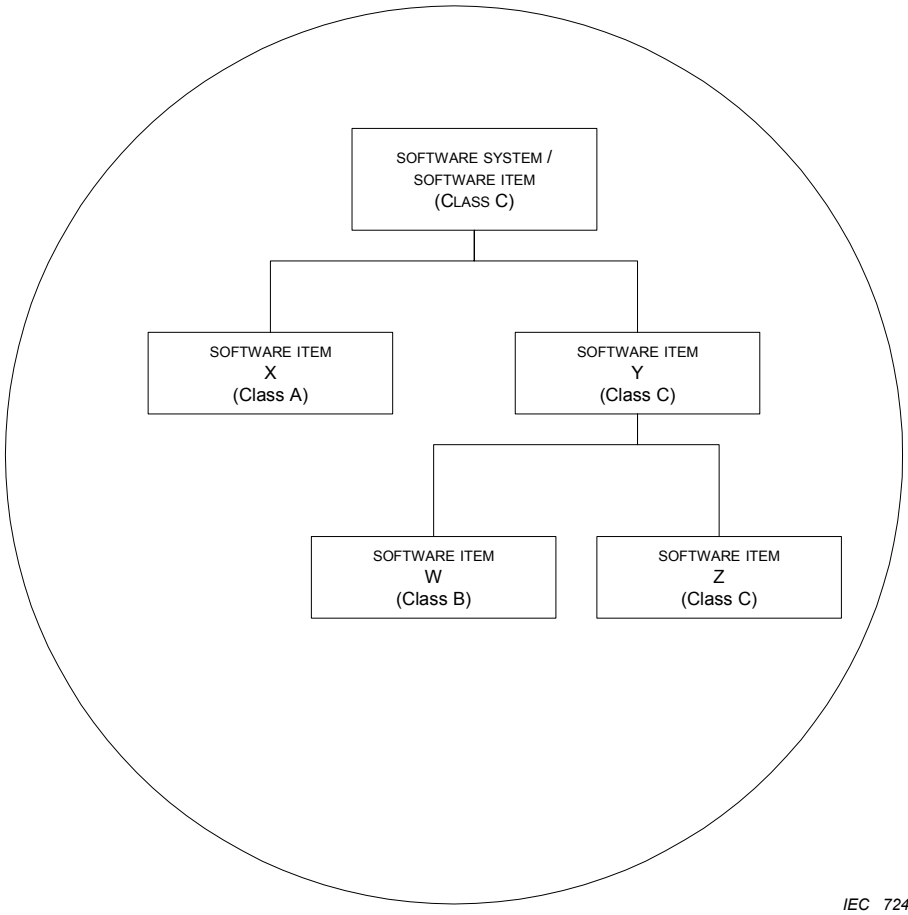

Figure B.1 – Example of partitioning of SOFTWARE ITEMS

For this example, the MANUFACTURER knows, due to the type of MEDICAL DEVICE software being developed, that the preliminary software safety classification for the SOFTWARE SYSTEM is software safety class C. During software ARCHITECTURE design the MANUFACTURER has decided to partition the SYSTEM, as shown, with 3 SOFTWARE ITEMS – X, W and Z. The MANUFACTURER is able to segregate all SOFTWARE SYSTEM contributions to HAZARDS which could result in death or SERIOUS INJURY to SOFTWARE ITEM Z and all remaining SOFTWARE SYSTEM contributions to HAZARDS which could result in a non-SERIOUS INJURY to SOFTWARE ITEM W. SOFTWARE ITEM W is classified as software safety class B and SOFTWARE ITEM Z is at software safety class C. SOFTWARE ITEM Y therefore must be classified as Class C, per 4.3 d). The SOFTWARE SYSTEM is also at a software safety class C per this requirement. SOFTWARE ITEM X has been classified at a software safety class of A. The MANUFACTURER is able to document a rationale for the segregation between SOFTWARE ITEMS X and Y, as well as SOFTWARE ITEMS W and Z, to assure the integrity of the segregation. If partitioning is not possible SOFTWARE ITEMS X and Y must be classified in software safety class C.

B.5 Software development PROCESS

B.5.1 Software development planning

The objective of this ACTIVITY is to plan the software development TASKS to reduce RISKS caused by software, communicate procedures and goals to members of the development team, and ensure that SYSTEM quality requirements for the MEDICAL DEVICE SOFTWARE are met.

The software development planning ACTIVITY can document TASKS in a single plan or in multiple plans. Some MANUFACTURERS might have established policies and procedures that apply to the development of all their MEDICAL DEVICE SOFTWARE. In this case the plan can simply reference the existing policies and procedures. Some MANUFACTURERS might prepare a plan or set of

de plans spécifique au développement de chaque PRODUIT LOGICIEL DE DISPOSITIF MÉDICAL qui reprenne en détails les ACTIVITÉS spécifiques et fasse référence aux procédures générales. Une autre possibilité est d'adapter un plan ou un ensemble de plans au développement de chaque PRODUIT LOGICIEL DE DISPOSITIF MÉDICAL. Il convient que la planification précise les niveaux de détail nécessaires pour réaliser le PROCESSUS de développement et soit proportionnelle au RISQUE encouru. Par exemple, les SYSTÈMES ou les éléments à RISQUE plus élevé pourraient faire l'objet d'un PROCESSUS de développement plus rigoureux et les TÂCHES pourraient être décrites de manière plus détaillée.

La planification est une ACTIVITÉ dynamique itérative qu'il convient de réexaminer et de remettre à jour au fur et à mesure du développement. Le plan peut évoluer pour intégrer davantage ou de meilleures informations au fur et à mesure de la compréhension du SYSTÈME et du niveau d'effort nécessaire au développement du SYSTÈME. Par exemple, la classification initiale de la sécurité du logiciel d'un SYSTÈME peut changer suite au PROCESSUS de GESTION DES RISQUEs et du développement de l'ARCHITECTURE du logiciel. Ou encore, il peut être décidé d'intégrer UN LOGICIEL SOUP aU SYSTÈME. Il est important que le(s) plan(s) soi(en)t mis à jour afin de refléter la connaissance courante du SYSTÈME et le niveau de rigueur requis du SYSTÈME ou des éléments du SYSTÈME pour permettre de maîtriser correctement le PROCESSUS de développement.

B.5.2 Analyses des exigences du logiciel

Cette ACTIVITÉ exige que le FABRICANT établisse et vérifie les exigences du logiciel pour le LOGICIEL DE DISPOSITIF MÉDICAL. L'établissement d'exigences vérifiables est essentiel pour déterminer ce qui doit être construit, pour déterminer que le LOGICIEL DE DISPOSITIF MÉDICAL présente un comportement acceptable et pour démontrer que le LOGICIEL DE DISPOSITIF MÉDICAL terminé est prêt pour utilisation. Pour démontrer que les exigences ont été mises en œuvre comme prévu, il convient que chaque exigence soit indiquée de sorte que les critères objectifs puissent être établis afin de déterminer si une mise en œuvre correcte a bien été faite. Si le PROCESSUS de GESTION DES RISQUES du dispositif impose au logiciel des exigences de MAÎTRISE DES RISQUES IDENTIFIÉS, ces exigences doivent être identifiées dans les exigences du logiciel de façon à ce qu'il soit possible d'assurer une TRAÇABILITÉ rattachant les mesures de MAÎTRISE DES RISQUES aux exigences du logiciel. Il est recommandé que toutes les exigences du logiciel soient identifiées de façon à permettre de démontrer la TRAÇABILITÉ entre l'exigence et les essais du SYSTÈME LOGICIEL. Si dans certains pays l'approbation d'un organisme de contrôle exige la conformité à des réglementations ou à des normes internationales spécifiques, il convient de documenter cette exigence de conformité dans les exigences de logiciel. Les exigences du logiciel établissent ce qui doit être mis en œuvre dans le logiciel et par conséquent, il est nécessaire de les évaluer avant de terminer l'ACTIVITÉ d'analyse des exigences.

Il est fréquent de confondre les besoins du client, les éléments d'entrée de conception, les exigences du logiciel, les spécifications fonctionnelles du logiciel et les spécifications de conception du logiciel. Les éléments d'entrée de la conception sont l'interprétation des besoins du client en exigences relatives au DISPOSITIF MÉDICAL, formellement documentées. Les exigences du logiciel sont les spécifications formellement documentées de ce que le logiciel réalise pour répondre aux besoins du client et de satisfaire aux éléments d'entrée de la conception. Les spécifications fonctionnelles du logiciel sont souvent incluses dans les exigences du logiciel et définissent en détails ce que le logiciel réalise pour satisfaire à ses exigences même s'il existe de nombreux autres moyens de satisfaire également aux exigences. Les spécifications de conception du logiciel définissent la manière dont le logiciel sera conçu et décomposé pour mettre en œuvre ses exigences et ses spécifications fonctionnelles.

Historiquement, les exigences du logiciel, les spécifications fonctionnelles et les spécifications de conception étaient écrites sous la forme d'un ensemble composé d'un ou de plusieurs documents. Il est aujourd'hui possible de considérer ces informations comme des éléments de données au sein d'une base de données commune. Chaque élément aurait alors un ou plusieurs attributs qui définiraient son objectif et ses liens avec d'autres éléments dans la base de données. Cette approche permet de présenter et d'imprimer différentes vues des informations qui correspondent le mieux à chaque ensemble d'utilisateurs destinataires

plans specific to the development of each MEDICAL DEVICE SOFTWARE PRODUCT that spell out in detail specific ACTIVITIES and reference general procedures. Another possibility is that a plan or set of plans is tailored for the development of each MEDICAL DEVICE SOFTWARE PRODUCT. The planning should be specified at the level of detail necessary to carry out the development PROCESS and should be proportional to the RISK. For example, SYSTEMS or items with higher RISK would be subject to a development PROCESS with more rigor and TASKS should be spelled out in greater detail.

Planning is an iterative ACTIVITY that should be re-examined and updated as development progresses. The plan can evolve to incorporate more and better information as more is understood about the SYSTEM and the level of effort needed to develop the SYSTEM. For example, a SYSTEM's initial software safety classification can change as a result of exercising the RISK MANAGEMENT PROCESS and development of the software ARCHITECTURE. Or it might be decided that a SOUP be incorporated into the SYSTEM. It is important that the plan(s) be updated to reflect current knowledge of the SYSTEM and the level of rigor needed for the SYSTEM or items in the SYSTEM to enable proper control over the development PROCESS.

B.5.2 Software requirements analysis

This ACTIVITY requires the MANUFACTURER to establish and verify the software requirements for the MEDICAL DEVICE SOFTWARE. Establishing verifiable requirements is essential for determining what is to be built, for determining that the MEDICAL DEVICE SOFTWARE exhibits acceptable behaviour, and for demonstrating that the completed MEDICAL DEVICE SOFTWARE is ready for use. To demonstrate that the requirements have been implemented as desired, each requirement should be stated in such a way that objective criteria can be established to determine whether it has been implemented correctly. If the device RISK MANAGEMENT PROCESS imposes requirements on the software to control identified RISKS, these requirements are to be identified in the software requirements in such a way as to make it possible to trace the RISK CONTROL measures to the software requirements. All software requirements should be identified in such a way as to make it possible to demonstrate TRACEABILITY between the requirement and SOFTWARE SYSTEM testing. If regulatory approval in some countries requires conformance to specific regulations or international standards, this conformance requirement should be documented in the software requirements. Because the software requirements establish what is to be implemented in the software, an evaluation of the requirements is required before the requirements analysis ACTIVITY is complete.

An area of frequent confusion is the distinction between customer needs, design inputs, software requirements, software functional specifications, and software design specifications. Design inputs are the interpretation of customer needs into formally documented MEDICAL DEVICE requirements. Software requirements are the formally documented specifications of what the software does to meet the customer needs and the design inputs. Software functional specifications are often included with the software requirements and define in detail what the software does to meet its requirements even though many different alternatives might also meet the requirements. Software design specifications define how the software will be designed and decomposed to implement its requirements and functional specifications.

Traditionally, software requirements, functional specifications, and design specifications have been written as a set of one or more documents. It is now feasible to consider this information as data items within a common database. Each item would have one or more attributes that would define its purpose and linkage to other items in the database. This approach allows presentation and printing of different views of the information best suited for each set of

(par exemple, marketing, FABRICANTS, contrôleurs, auditeurs); cette approche prend également en charge la TRAÇABILITÉ ce qui permet de démontrer que la mise en œuvre a été correctement réalisée et dans quelle mesure les jeux d'essais permettent de vérifier les exigences. Les outils qui prennent en charge cette approche peuvent être tout simplement des documents hypertext utilisant des hyperliens HTML ou aussi complexes et fonctionnels que des outils d'ingénierie logicielle assistée par ordinateur (en anglais, CASE Computer Aided Software Engineering).

Le PROCESSUS d'exigences SYSTÈME n'est pas l'objet de la présente norme. Cependant, la décision de mettre en œuvre avec le logiciel une fonctionnalité DE DISPOSITIF MÉDICAL est normalement prise lors de la conception du SYSTÈME. Certaines ou toutes les exigences du SYSTÈME font l'objet d'une affectation pour être mises en œuvre dans le logiciel. L'ACTIVITÉ d'analyse des exigences du logiciel consiste à analyser les exigences attribuées au logiciel par le PROCESSUS des exigences SYSTÈME et à en extraire un ensemble exhaustif d'exigences logicielles qui reflètent les exigences qui lui sont attribuées.

Pour assurer l'intégrité du SYSTÈME, il convient que le FABRICANT prévoit un mécanisme de prise en charge des modifications et clarifications apportées aux exigences du SYSTÈME afin de corriger les impossibilités, les incohérences ou les ambiguïtés, que ce soit dans les exigences du SYSTÈME d'origine ou celles du logiciel.

Le PROCESSUS de collecte et d'analyse des exigences du SYSTÈME et du logiciel peut être itératif. La présente norme n'exige pas que les PROCESSUS soient séparés en deux couches strictement délimitées. Dans la pratique, l'ARCHITECTURE DU SYSTÈME et l'ARCHITECTURE du logiciel sont souvent décrites simultanément et leurs exigences respectives sont ensuite consignées par écrit sous forme de couche.

B.5.3 Conception ARCHITECTURALE du logiciel

Cette ACTIVITÉ exige que le FABRICANT définisse les principaux composants structurels du logiciel, leurs caractéristiques visibles de l'extérieur, ainsi que les relations qui existent entre eux. Si le comportement d'un composant peut affecter d'autres composants, il convient que ce comportement soit décrit dans l'ARCHITECTURE logicielle. Cette description est notamment importante lorsque le comportement peut affecter des composants du DISPOSITIF MÉDICAL extérieurs au logiciel. Les décisions relatives à l'ARCHITECTURE sont extrêmement importantes pour la mise en œuvre des mesures de MAÎTRISE DU RISQUE. Si on ne comprend pas (et si l'on ne documente pas) le comportement d'un composant qui peut affecter d'autres composants, il sera presque impossible de démontrer que le SYSTÈME est sûr. Une ARCHITECTURE logicielle est nécessaire pour s'assurer de la mise en œuvre correcte des exigences du logiciel. L'ARCHITECTURE du logiciel n'est complète que si toutes les exigences logicielles peuvent être mises en œuvre par les ÉLÉMENTS LOGICIELS identifiés. La conception et la mise en œuvre du logiciel étant dépendantes de l'ARCHITECTURE, l'ARCHITECTURE est vérifiée pour terminer cette ACTIVITÉ. Une ÉVALUATION technique est généralement utilisée pour la VÉRIFICATION de l'ARCHITECTURE.

La classification des ÉLÉMENTS LOGICIELS pendant l'ACTIVITÉ d'ARCHITECTURE du logiciel génère une base qui permet par la suite de choisir des PROCESSUS logiciels. Les enregistrements de cette classification sont mis en MAÎTRISE DES MODIFICATIONS comme partie intégrante du DOSSIER DE GESTION DES RISQUES.

La classification peut être invalidée par de nombreux événements ultérieurs, tels que, par exemple:

– les modifications de la spécification du SYSTÈME, de la spécification ou de l'ARCHITECTURE du logiciel;
– la découverte d'erreurs dans l'ANALYSE DU RISQUE, notamment en ce qui concerne des DANGERS non prévus; et
– la découverte de l'infaisabilité d'une exigence, notamment une mesure de MAÎTRISE DU RISQUE.

intended users (e.g., marketing, MANUFACTURERS, testers, auditors) and supports TRACEABILITY to demonstrate adequate implementation and the extent to which test cases test the requirements. Tools to support this approach can be as simple as a hypertext document using HTML hyperlinks or as complex and capable as computer aided software engineering (CASE) tools and requirements analysis tools.

The SYSTEM requirements PROCESS is out of scope of this standard. However, the decision to implement MEDICAL DEVICE functionality with software is normally made during SYSTEM design. Some or all of the SYSTEM requirements are allocated to be implemented in software. The software requirements analysis ACTIVITY consists of analyzing the requirements allocated to software by the SYSTEM requirements PROCESS and deriving a comprehensive set of software requirements that reflect the allocated requirements.

To ensure the integrity of the SYSTEM, the MANUFACTURER should provide a mechanism for negotiating changes and clarifications to the SYSTEM requirements to correct impracticalities, inconsistencies or ambiguities in either the parent SYSTEM requirements or the software requirements.

The PROCESS of capture and analysis of SYSTEM and software requirements can be iterative. This standard does not intend to require the PROCESSES to be rigidly segregated into two layers. In practice, SYSTEM ARCHITECTURE and software ARCHITECTURE are often outlined simultaneously and the SYSTEM and software requirements are subsequently documented in a layered form.

B.5.3 Software ARCHITECTURAL design

This ACTIVITY requires the MANUFACTURER to define the major structural components of the software, their externally visible properties, and the relationship among them. If the behaviour of a component can affect other components, that behavior should be described in the software ARCHITECTURE. This description is especially important for behaviour that can affect components of the MEDICAL DEVICE that are outside the software. ARCHITECTURAL decisions are extremely important for implementing RISK CONTROL measures. Without understanding (and documenting) the behaviour of a component that can affect other components, it will be nearly impossible to show that the SYSTEM is safe. A software ARCHITECTURE is necessary to ensure the correct implementation of the software requirements. The software ARCHITECTURE is not complete unless all software requirements can be implemented by the identified SOFTWARE ITEMS. Because the design and implementation of the software is dependent on the ARCHITECTURE, the ARCHITECTURE is VERIFIED to complete this ACTIVITY. VERIFICATION of the ARCHITECTURE is generally done by technical EVALUATION.

The classification of SOFTWARE ITEMS during the software ARCHITECTURE ACTIVITY creates a basis for the subsequent choice of software PROCESSES. The records of classification are placed under change control as part of the RISK MANAGEMENT FILE.

Many subsequent events might invalidate the classification. These include, for example:

– changes of SYSTEM specification, software specification or ARCHITECTURE;
– discovery of errors in the RISK ANALYSIS, especially unforeseen HAZARDS; and
– discovery of the infeasibility of a requirement, especially a RISK CONTROL measure;

Par conséquent, pendant toutes les ACTIVITÉS faisant suite à la conception de l'ARCHITECTURE du logiciel, il convient de réévaluer la classification du SYSTÈME LOGICIEL et des ÉLÉMENTS LOGICIELS et peut-être de la réviser. Ceci déclencherait une reprise du travail POUR L'APPLICATION DE PROCESSUS supplémentaires à un ÉLÉMENT LOGICIEL suite à son passage dans une classe supérieure. Le PROCESSUS de gestion de la configuration du logiciel (Article 8) est utilisé pour s'assurer que toute retouche nécessaire a été identifiée et réalisée.

B.5.4 Conception détaillée du logiciel

Cette ACTIVITÉ exige que le FABRICANT décompose les ÉLÉMENTS LOGICIELS et interfaces définis dans l'ARCHITECTURE pour créer des UNITÉS LOGICIELLES et leurs interfaces. Même si les UNITÉS LOGICIELLES sont souvent considérées comme étant une fonction ou un module simple, cette opinion n'est pas toujours valable. Nous avons défini l'UNITÉ LOGICIELLE comme étant un ÉLÉMENT LOGICIEL qui n'est pas subdivisé en éléments plus petits. Les UNITÉS LOGICIELLES peuvent être soumises à des essais séparément. Il convient que le FABRICANT définisse le niveau de détails de l'UNITÉ LOGICIELLE. La conception détaillée spécifie des algorithmes, des représentations des données, des interfaces entre les différentes UNITÉS LOGICIELLES et les interfaces entre les UNITÉS LOGICIELLES et les structures de données. La conception détaillée doit également traiter du conditionnement (programmation) du PRODUIT LOGICIEL. Il est nécessaire de documenter la conception de chaque UNITÉ LOGICIELLE et de ses interfaces de manière à pouvoir correctement la mettre en œuvre. La conception détaillée renseigne les détails nécessaires à la construction du logiciel. Il convient qu'elle soit suffisamment complète pour que le programmateur n'ait pas à prendre des décisions de conception circonstancielles.

Un ÉLÉMENT LOGICIEL peut être décomposé de sorte que seul un nombre infime de nouveaux ÉLÉMENTS LOGICIELS mette en œuvre l'exigence relative à la SÉCURITÉ de l'ÉLÉMENT LOGICIEL d'origine. Les autres ÉLÉMENTS LOGICIELS ne mettent pas en œuvre des fonctions relatives à la SÉCURITÉ et peuvent être reclassés dans une classe inférieure de sécurité logicielle. Cependant, cette décision fait en elle-même partie du PROCESSUS DE GESTION DES RISQUES et elle est consignée dans le DOSSIER DE GESTION DES RISQUES.

Etant donné qu'une mise en œuvre dépend d'une conception détaillée, il est nécessaire de vérifier la conception détaillée avant de terminer l'ACTIVITÉ. Une ÉVALUATION technique est généralement effectuée pour la VÉRIFICATION de la conception détaillée. Le paragraphe 5.4.4 exige que le fabricant vérifie les éléments de sortie des activités de conception détaillée. La conception spécifie la manière dont les exigences doivent être mises en œuvre. Si la conception comporte des défauts, le code ne mettra pas correctement les exigences en application.

Il convient que le fabricant vérifie les caractéristiques de la conception qu'il estime importantes pour la SÉCURITÉ, s'il en existe. Les exemples de ces caractéristiques comprennent:

– la mise en œuvre des évènements prévus, les entrées, les sorties, les interfaces, le logigramme, l'allocation du processeur principal (CPU), l'allocation des ressources mémoire, les définitions de l'erreur et de l'exception, l'isolement de l'erreur et de l'exception, et la reprise sur erreur;
– la définition de l'état défectueux dans lequel toutes les fautes qui peuvent entraîner une situation dangereuse sont traitées, avec les évènements et les transitions;
– l'initialisation des variables, la gestion de la mémoire; et
– les réinitialisations à chaud et à froid, la mis en veille et les autres changements d'état qui peuvent affecter les mesures de MAÎTRISE DU RISQUE.

B.5.5 Mise en œuvre et vérification de l'UNITÉ LOGICIELLE

Cette ACTIVITÉ exige que le FABRICANT écrive et vérifie le code des UNITÉS LOGICIELLES. La conception détaillée doit être traduite en code source. Le codage représente le point où s'achève la décomposition des spécifications et où commence la composition du logiciel exécutable. Pour assurer la cohérence des caractéristiques souhaitables du code, il convient

Therefore, during all ACTIVITIES following the design of the software ARCHITECTURE, the classification of the SOFTWARE SYSTEM and SOFTWARE ITEMS should be re-EVALUATED and might need to be revised. This would trigger rework to apply additional PROCESSES to a SOFTWARE ITEM as a result of its upgrading to a higher class. The software configuration management PROCESS (Clause 8) is used to ensure that all necessary rework is identified and completed.

B.5.4 Software detailed design

This ACTIVITY requires the MANUFACTURER to refine the SOFTWARE ITEMS and interfaces defined in the ARCHITECTURE to create SOFTWARE UNITS and their interfaces. Although SOFTWARE UNITS are often thought of as being a single function or module, this view is not always appropriate. We have defined SOFTWARE UNIT to be a SOFTWARE ITEM that is not subdivided into smaller items. SOFTWARE UNITS can be tested separately. The MANUFACTURER should define the level of detail of the SOFTWARE UNIT. Detailed design specifies algorithms, data representations, interfaces among different SOFTWARE UNITS, and interfaces between SOFTWARE UNITS and data structures. Detailed design must also be concerned with the packaging of the SOFTWARE PRODUCT. It is necessary to document the design of each SOFTWARE UNIT and its interface so that the SOFTWARE UNIT can be implemented correctly. The detailed design fills in the details necessary to construct the software. It should be complete enough that the programmer is not required to make ad hoc design decisions.

A SOFTWARE ITEM can be decomposed so that only a few of the new SOFTWARE ITEMS implement the SAFETY-related requirement of the original SOFTWARE ITEM. The remaining SOFTWARE ITEMS do not implement SAFETY-related functions and can be reclassified into a lower software safety class. However, the decision to do this is in itself part of the RISK MANAGEMENT PROCESS, and is documented in the RISK MANAGEMENT FILE.

Because implementation depends on detailed design, it is necessary to verify the detailed design before the ACTIVITY is complete. VERIFICATION of detailed design is generally done by a technical EVALUATION. Subclause 5.4.4 requires the MANUFACTURER to verify the outputs of the detailed design ACTIVITIES. The design specifies how the requirements are to be implemented. If the design contains defects, the code will not implement the requirements correctly.

When present in the design, the MANUFACTURER should verify design characteristics which the MANUFACTURER believes are important for SAFETY. Examples of these characteristics include:

– implementation of the intended events, inputs, outputs, interfaces, logic flow, allocation of CPU, allocation of memory resources, error and exception definition, error and exception isolation, and error recovery;
– definition of the default state, in which all faults that can result in a hazardous situation are addressed, with events and transitions;
– initialization of variables, memory management; and
– cold and warm resets, standby, and other state changes that can affect the RISK CONTROL measures.

B.5.5 SOFTWARE UNIT implementation and verification

This ACTIVITY requires the MANUFACTURER to write and verify the code for the SOFTWARE UNITS. The detailed design is to be translated into source code. Coding represents the point where decomposition of the specifications ends and composition of the executable software begins. To consistently achieve the desirable code characteristics, coding standards should be used to

d'utiliser des normes de codage afin de prescrire un style de codage préférentiel. Les normes de codage comprennent par exemple, des exigences d'intelligibilité, des règles ou des restrictions d'usage du langage et une gestion de la complexité. Pour chaque unité, le code est vérifié afin de s'assurer qu'il fonctionne comme spécifié dans la conception détaillée et qu'il est conforme aux normes de codage spécifiées.

Le paragraphe 5.5.5 exige que le FABRICANT vérifie le code. Si le code ne met pas correctement en application la conception, les performances du logiciel de DISPOSITIF MÉDICAL ne seront pas celles attendues.

B.5.6 Intégration et essai d'intégration du logiciel

Cette ACTIVITÉ exige que le FABRICANT planifie et exécute l'intégration des UNITÉS LOGICIELLES dans des ÉLÉMENTS LOGICIELS d'ensemble ainsi que l'intégration des ÉLÉMENTS LOGICIELS dans des ÉLÉMENTS LOGICIELS d'ensemble plus élevés, et qu'il vérifie que les ÉLÉMENTS LOGICIELS qui en résultent se comportent comme prévu.

Les approches correspondantes peuvent aller d'une intégration non incrémentielle à toute forme d'intégration incrémentielle. Les caractéristiques de l'ÉLÉMENT LOGICIEL à assembler imposent la méthode d'intégration choisie.

Les essais d'intégration de logiciel portent sur le transfert des données et leur contrôle entre les interfaces internes et externes d'un ÉLÉMENT LOGICIEL. Les interfaces externes sont celles que l'ÉLÉMENT LOGICIEL partage avec d'autres logiciels, y compris le logiciel du système d'exploitation ainsi qu'avec le matériel du DISPOSITIF MÉDICAL.

Il convient que la rigueur des essais d'intégration et le niveau de détail de la documentation associée aux essais d'intégration soient à la mesure du RISQUE associé au dispositif, qu'ils correspondent à la dépendance du dispositif vis-à-vis du logiciel pour les fonctions potentiellement DANGEREUSES et qu'ils tiennent compte du rôle que jouent les ÉLÉMENTS LOGICIELS spécifiques dans des fonctions à haut RISQUE du DISPOSITIF MÉDICAL. Par exemple, même s'il est recommandé de soumettre aux essais tous les ÉLÉMENTS LOGICIELS, il convient de soumettre ceux qui ont un effet sur la SÉCURITÉ à des essais plus directs, plus approfondis et plus détaillés.

Le cas échéant, les essais d'intégration démontrent le comportement du programme aux limites de ses domaines d'entrée et de sortie et confirment les réponses du programme à des éléments d'entrée non valables, inattendus et particuliers. Les actions du programme sont révélées lorsqu'elles reçoivent des combinaisons d'éléments d'entrée ou des séquences d'éléments d'entrée inattendues ou lorsque des exigences de temporisation bien définies sont violées. Si nécessaire, il convient d'inclure dans les exigences d'essai du plan, les essais de type «boîte blanche» à réaliser dans le cadre des essais d'intégration.

Les essais de «boîte blanche», également appelés *essais transparents*, *structurels, de «boîte claire» et de «boîte ouverte»* sont une technique d'essai pour sélectionner les données d'essai qui utilise une connaissance explicite du fonctionnement interne de l'ÉLÉMENT LOGICIEL soumis aux essais. Les essais de «boîte blanche» utilisent une connaissance spécifique de l'ÉLÉMENT LOGICIEL pour examiner ces résultats en sortie. L'essai n'est précis que si le contrôleur sait ce qu'est supposé faire l'ÉLÉMENT LOGICIEL. Le contrôleur peut alors voir si l'ÉLÉMENT LOGICIEL déroge à son objectif prévu. Les essais de «boîte blanche» ne peuvent garantir que la spécification dans son ensemble a été mise en œuvre car ils se concentrent sur des essais de mise en œuvre de l'ÉLÉMENT LOGICIEL. Les essais de «boîte noire» également appelés essais comportementaux, fonctionnels, de «boîte opaque» et de «boîte fermée» vérifient la spécification fonctionnelle et il n'est pas possible de garantir que toutes les parties de la mise en œuvre ont été contrôlées. Ainsi, les essais de «boîte noire» ont pour base la spécification et décèleront des omissions indiquant que telle partie de la spécification n'a pas été remplie. Les essais de «boîte blanche» se fondent sur la mise en œuvre et décèleront des commandes indiquant que telle partie de la mise en œuvre est défectueuse. Pour contrôler pleinement un PRODUIT LOGICIEL, des essais de «boîte noire» et des essais de «boîte blanche» pourraient être exigés.

specify a preferred coding style. Examples of coding standards include requirements for understandability, language usage rules or restrictions, and complexity management. The code for each unit is VERIFIED to ensure that it functions as specified by the detailed design and that it complies with the specified coding standards.

Subclause 5.5.5 requires the MANUFACTURER to verify the code. If the code does not implement the design correctly, the MEDICAL DEVICE SOFTWARE will not perform as intended.

B.5.6 Software integration and integration testing

This ACTIVITY requires the MANUFACTURER to plan and execute integration of SOFTWARE UNITS into aggregate SOFTWARE ITEMS as well as integration of SOFTWARE ITEMS into higher aggregated SOFTWARE ITEMS and to verify that the resulting SOFTWARE ITEMS behave as intended.

The approach to integration can range from non-incremental integration to any form of incremental integration. The properties of the SOFTWARE ITEM being assembled dictate the chosen method of integration.

Software integration testing focuses on the transfer of data and control across a SOFTWARE ITEM's internal and external interfaces. External interfaces are those with other software, including operating system software, and MEDICAL DEVICE hardware.

The rigor of integration testing and the level of detail of the documentation associated with integration testing should be commensurate with the RISK associated with the device, the device's dependence on software for potentially hazardous functions, and the role of specific SOFTWARE ITEMS in higher RISK device functions. For example, although all SOFTWARE ITEMS should be tested, items that have an effect on SAFETY should be subject to more direct, thorough, and detailed tests.

As applicable, integration testing demonstrates program behaviour at the boundaries of its input and output domains and confirms program responses to invalid, unexpected, and special inputs. The program's actions are revealed when given combinations of inputs or unexpected sequences of inputs, or when defined timing requirements are violated. The test requirements in the plan should include, as appropriate, the types of white box testing to be performed as part of integration testing.

White box testing, also known as *glass box*, *structural*, *clear box* and *open box testing*, is a testing technique where explicit knowledge of the internal workings of the SOFTWARE ITEM being tested are used to select the test data. White box testing uses specific knowledge of the SOFTWARE ITEM to examine outputs. The test is accurate only if the tester knows what the SOFTWARE ITEM is supposed to do. The tester can then see if the SOFTWARE ITEM diverges from its intended goal. White box testing cannot guarantee that the complete specification has been implemented since it is focused on testing the implementation of the SOFTWARE ITEM. Black box testing, also known as behavioural, functional, opaque-box, and closed-box testing, is focused on testing the functional specification and it cannot guarantee that all parts of the implementation have been tested. Thus black box testing is testing against the specification and will discover faults of omission, indicating that part of the specification has not been fulfilled. White box testing is testing against the implementation and will discover faults of commission, indicating that part of the implementation is faulty. In order to fully test a SOFTWARE PRODUCT both black and white box testing might be required.

Les plans et la documentation d'essai identifiés en 5.6 et 5.7 peuvent être des documents individuels liés à des phases spécifiques de développement ou de prototypes évolutifs. Ils pourraient également être associés de telle sorte qu'un seul document ou ensemble de documents couvre les exigences de sous-sections multiples. Tout ou partie de ces documents pourrait être incorporé dans des documents de projet de niveau supérieur, tels que le plan d'assurance qualité du logiciel ou du projet ou un plan d'essai global qui traite de tous les aspects des essais du matériel et du logiciel. Dans ce cas, il convient de créer un renvoi qui identifie la manière dont les divers documents de projet sont corrélés à chacune des TÂCHES d'intégration du logiciel.

Les essais d'intégration du logiciel peuvent être réalisés dans un environnement simulé, sur un matériel cible réel ou sur le DISPOSITIF MÉDICAL complet.

Le paragraphe 5.6.2 exige que le FABRICANT vérifie les éléments de sortie de l'ACTIVITÉ d'intégration du logiciel. Le résultat de l'ACTIVITÉ d'intégration de logiciel est représenté par les ÉLÉMENTS LOGICIELS intégrés. Ces ÉLÉMENTS LOGICIELS intégrés doivent fonctionner correctement pour que l'ensemble du LOGICIEL DE DISPOSITIF MÉDICAL fonctionne correctement et en toute SÉCURITÉ.

B.5.7 Essais du SYSTÈME LOGICIEL

Cette ACTIVITÉ exige que le FABRICANT vérifie la fonctionnalité du logiciel en s'assurant que les exigences correspondantes ont été mises en œuvre avec succès.

Les essais du SYSTÈME LOGICIEL démontrent que la fonctionnalité spécifiée existe. Ces essais vérifient la fonctionnalité et les performances du programme tel que construit, par rapport aux exigences du logiciel.

Les essais du SYSTÈME LOGICIEL portent sur les essais fonctionnels (boîte noire), bien qu'il pourrait être souhaitable d'utiliser des méthodes «boîte blanche» (voir paragraphe précédent) pour plus d'efficacité de certains essais, en initiant des conditions de contraintes ou des défauts, ou en augmentant la portée des essais de qualification du code. L'organisation des essais par types et étapes est flexible, il convient cependant de démontrer et documenter la couverture des exigences de la MAÎTRISE DU RISQUE, de l'aptitude à l'usage et des types d'essais (par exemple, défauts, installations, contraintes).

Les essais du SYSTÈME LOGICIEL vérifient le logiciel intégré et peuvent être effectués dans un environnement simulé, sur un matériel cible réel ou sur le DISPOSITIF MÉDICAL complet.

Lorsqu'une modification est apportée à un SYSTÈME LOGICIEL (même si elle est infime), il convient de déterminer le niveau des essais de régression (et non uniquement les essais de la modification particulière) pour s'assurer qu'aucun effet secondaire imprévu n'a été introduit. Il convient de planifier et de documenter ces essais de régression (en indiquant les raisons pour lesquelles les essais du SYSTÈME LOGICIEL n'ont pas été totalement recommencés).

Les responsabilités relatives aux essais du SYSTÈME LOGICIEL peuvent être dispersées en plusieurs lieux et confiées à différentes organisations. Cependant, quelle que soit la répartition des TÂCHES, les rapports contractuels, les sources de composants ou l'environnement de développement, le FABRICANT du dispositif conserve la responsabilité finale et doit s'assurer que le logiciel fonctionne correctement pour son usage prévu.

Si des ANOMALIEs non décelées au cours des essais sont récurrentes mais qu'il a été décidé de ne pas les corriger, il est nécessaire d'ÉVALUER ces ANOMALIES au titre de l'analyse des DANGERS afin de vérifier qu'elles n'affectent pas la SÉCURITÉ du dispositif. Il convient de comprendre les causes inhérentes et les symptômes des ANOMALIES et de documenter les raisons pour lesquelles elles n'ont pas été corrigées.

The plans and test documentation identified in 5.6 and 5.7 can be individual documents tied to specific phases of development or evolutionary prototypes. They also might be combined so a single document or set of documents covers the requirements of multiple subsections. All or portions of the documents could be incorporated into higher level project documents such as a software or project quality assurance plan or a comprehensive test plan that addresses all aspects of testing for hardware and software. In these cases, a cross reference should be created that identifies how the various project documents relate to each of the software integration TASKS.

Software integration testing can be performed in a simulated environment, on actual target hardware, or on the full MEDICAL DEVICE.

Subclause 5.6.2 requires the MANUFACTURER to verify the output of the software integration ACTIVITY. The output of the software integration ACTIVITY is the integrated SOFTWARE ITEMS. These integrated SOFTWARE ITEMS must function properly for the entire MEDICAL DEVICE SOFTWARE to function correctly and safely.

B.5.7 SOFTWARE SYSTEM testing

This ACTIVITY requires the MANUFACTURER to verify the software's functionality by verifying that the requirements for the software have been successfully implemented.

SOFTWARE SYSTEM testing demonstrates that the specified functionality exists. This testing VERIFIES the functionality and performance of the program as built with respect to the requirements for the software.

SOFTWARE SYSTEM testing focuses on functional (black box) testing, although it might be desirable to use white box (see previous section) methods to more efficiently accomplish certain tests, initiate stress conditions or faults, or increase code coverage of the qualification tests. The organization of testing by types and test stage is flexible, but coverage of requirements, RISK CONTROL, usability, and test types (e.g., fault, installation, stress) should be demonstrated and documented.

SOFTWARE SYSTEM testing tests the integrated software and can be performed in a simulated environment, on actual target hardware, or on the full MEDICAL DEVICE.

When a change is made to a SOFTWARE SYSTEM (even a small change), the degree of REGRESSION TESTING (not just the testing of the individual change) should be determined to ensure that no unintended side effects have been introduced. This REGRESSION TESTING (and the rationale for not fully repeating SOFTWARE SYSTEM testing) should be planned and documented.

SOFTWARE SYSTEM test responsibilities can be dispersed, occurring at different locations and being conducted by different organizations. However, regardless of the distribution of TASKS, contractual relations, source of components, or development environment, the device MANUFACTURER retains ultimate responsibility for ensuring that the software functions properly for its intended use.

If ANOMALIES uncovered during testing can be repeated, but a decision has been made not to fix them, then these ANOMALIES need to be EVALUATED in relation to the HAZARD analysis to verify that they do not affect the SAFETY of the device. The root cause and symptoms of the ANOMALIES should be understood, and the rationale for not fixing them should be documented.

Le paragraphe 5.7.4 exige que les résultats des essais du SYSTÈME LOGICIEL soient ÉVALUÉS pour s'assurer que les résultats prévus sont bien obtenus.

B.5.8 Diffusion du logiciel

Cette ACTIVITÉ exige que le FABRICANT documente la VERSION diffusée du LOGICIEL DE DISPOSITIF MÉDICAL, qu'il précise comment elle a été créée et qu'il se conforme à des procédures appropriées de diffusion du logiciel.

Il convient que le FABRICANT soit capable de démontrer que le logiciel qui a été développé sur la base du PROCESSUS de développement est bien le logiciel qui est diffusé. Si nécessaire, il convient que le FABRICANT soit capable de récupérer le logiciel et les outils utilisés pour le générer et il est recommandé que l'entreposage, le conditionnement et la livraison du logiciel réduisent au minimum les RISQUES de DOMMAGES ou d'usage abusif. Il convient d'établir des procédures bien définies pour s'assurer que ces TÂCHES sont réalisées correctement et qu'elles donnent des résultats cohérents.

B.6 PROCESSUS de maintenance du logiciel

B.6.1 Etablissement du plan de maintenance du logiciel

LE PROCESSUS de maintenance du logiciel diffère, par rapport au PROCESSUS de développement du logiciel, de deux manières:

– Le FABRICANT est autorisé à utiliser un PROCESSUS moins étendu que le PROCESSUS de développement complet du logiciel pour mettre en œuvre des modifications rapides en réponse à des problèmes urgents.
– Lorsqu'il répond au RAPPORT DE PROBLÈMES du logiciel, concernant le produit diffusé, le FABRICANT traite non seulement des problèmes mais doit également satisfaire aux réglementations locales (en général, en mettant en place un plan de surveillance proactif pour le recueil des données du problème sur le terrain et pour communiquer avec les utilisateurs et les autorités réglementaires au sujet du problème).

Le paragraphe 6.1 exige que ces PROCESSUS soient définis dans un plan de maintenance.

Cette ACTIVITÉ exige que le FABRICANT élabore ou identifie des procédures de mise en œuvre des ACTIVITÉS et TÂCHES de maintenance. Pour la mise en œuvre des actions correctives, la MAÎTRISE DES MODIFICATIONS en maintenance et la gestion de la diffusion du logiciel révisé, il convient que le FABRICANT consigne et résolve les rapports de problèmes et les demandes des utilisateurs tout en gérant les modifications apportées au LOGICIEL DE DISPOSITIF MÉDICAL. Ce PROCESSUS est activé lorsque le LOGICIEL DE DISPOSITIF MÉDICAL subit des modifications touchant au code et à la documentation correspondante du fait d'un problème ou d'une nécessité d'amélioration ou d'adaptation. L'objectif est de modifier le LOGICIEL DE DISPOSITIF MÉDICAL diffusé tout en préservant son intégrité. Ce PROCESSUS couvre également la migration du LOGICIEL DE DISPOSITIF MÉDICAL vers des environnements ou des plates-formes pour lesquels il n'était pas initialement diffusé. Les ACTIVITÉS prévues dans cet article sont spécifiques au PROCESSUS de maintenance; cependant, le PROCESSUS de maintenance pourrait utiliser d'autres PROCESSUS de la présente norme.

Il est nécessaire que le FABRICANT planifie la manière dont les ACTIVITÉS et LES TÂCHES du PROCESSUS de maintenance seront réalisées.

B.6.2 Analyse des problèmes et des modifications

Cette ACTIVITÉ exige que le FABRICANT analyse le retour d'information pour son effet; qu'il vérifie les problèmes qui sont rapportés; et qu'il envisage, choisisse et obtienne l'approbation de la mise en œuvre d'une option de modification donnée. Les problèmes et les autres DEMANDES DE MODIFICATION peuvent affecter les performances, la SÉCURITÉ ou les autorisations réglementaires d'un DISPOSITIF MÉDICAL. Une analyse est nécessaire pour déterminer s'il existe

Subclause 5.7.4 requires the results of the SOFTWARE SYSTEM testing be EVALUATED to ensure that the expected results were obtained.

B.5.8 Software release

This ACTIVITY requires the MANUFACTURER to document the VERSION of the MEDICAL DEVICE SOFTWARE being released, specify how it was created, and follow appropriate procedures for release of the software.

The MANUFACTURER should be able to show that the software that was developed using the development PROCESS is the software that is being released. The MANUFACTURER should also be able to retrieve the software and the tools used for its generation in case it is needed in the future and should store, package, and deliver the software in a manner that minimizes the software from being damaged or misused. Defined procedures should be established to ensure that these TASKS are performed appropriately and with consistent results.

B.6 Software maintenance PROCESS

B.6.1 Establish software maintenance plan

The software maintenance PROCESS differs from the software development PROCESS in two ways:

– The MANUFACTURER is permitted to use a smaller PROCESS than the full software development PROCESS to implement rapid changes in response to urgent problems.

– In responding to software PROBLEMS REPORTS relating to released product, the MANUFACTURER not only addresses the problem but also satisfies local regulations (typically by running a pro-active surveillance scheme for collecting problem data from the field and communicating with users and regulators about the problem).

Subclause 6.1 requires these PROCESSES to be established in a maintenance plan.

This ACTIVITY requires the MANUFACTURER to create or identify procedures for implementing maintenance ACTIVITIES and TASKS. To implement corrective actions, control changes during maintenance, and manage release of revised software, the MANUFACTURER should document and resolve reported problems and requests from users, as well as manage modifications to the MEDICAL DEVICE SOFTWARE. This PROCESS is activated when the MEDICAL DEVICE SOFTWARE undergoes modifications to code and associated documentation because of either a problem or the need for improvement or adaptation. The objective is to modify released MEDICAL DEVICE SOFTWARE while preserving its integrity. This PROCESS includes migration of the MEDICAL DEVICE SOFTWARE to environments or platforms for which it was not originally released. The ACTIVITIES provided in this clause are specific to the maintenance PROCESS; however, the maintenance PROCESS might use other PROCESSES in this standard.

The MANUFACTURER needs to plan how the ACTIVITIES and TASKS of the maintenance PROCESS will be performed.

B.6.2 Problem and modification analysis

This ACTIVITY requires the MANUFACTURER to analyze feedback for its effect; verify reported problems; and consider, select, and obtain approval for implementing a modification option. Problems and other requests for changes can affect the performance, SAFETY, or regulatory clearance of a MEDICAL DEVICE. An analysis is necessary to determine whether any effects exist

d'éventuels effets induits par un RAPPORT DE PROBLÈME ou si ces effets nécessiteront une modification pour corriger un problème ou répondre à une demande. Il est notamment important de vérifier, par analyse de TRAÇABILITÉ ou de régression, que les mesures de MAÎTRISE DU RISQUE intégrées au dispositif ne sont pas altérées ou modifiées de manière préjudiciable par la modification logicielle mise en œuvre dans le cadre de l'ACTIVITÉ de maintenance du logiciel. Il est également important de vérifier que le logiciel modifié n'entraîne pas un DANGER ou n'occulte pas un RISQUE alors que précédemment il n'entraînait pas de DANGER ou n'occultait pas de RISQUES dans le logiciel. La classification de SÉCURITÉ d'un ÉLÉMENT LOGICIEL pourrait avoir changé si la modification du logiciel entraîne à présent un DANGER ou occulte un RISQUE.

Il est important de faire la distinction entre maintenance du logiciel (Article 6) et résolution des problèmes de logiciel (Article 9).

Le PROCESSUS de maintenance du logiciel est axé sur l'apport d'une réponse appropriée aux retours d'information après diffusion du PRODUIT LOGICIEL. Le PROCESSUS de maintenance de logiciel, en tant que partie intégrante d'un DISPOSITIF MÉDICAL, doit s'assurer que:

– les RAPPORTS DE PROBLÈMES relatifs à la sécurité sont traités et notifiés aux autorités compétentes de réglementation et aux utilisateurs concernés;
– les PRODUITS LOGICIELS sont revalidés et rediffusés après modification avec des contrôles formels qui permettent de s'assurer que le problème a été rectifié et que les problèmes ultérieurs sont évités;
– le FABRICANT vérifie si d'autres produits logiciels auraient pu être affectés et prend les mesures nécessaires.

La résolution des problèmes de logiciel est axée sur le fonctionnement d'un système de contrôle global qui:

- analyse LES RAPPORTS DE PROBLÈME et identifie l'ensemble des implications du problème;
- décide d'effectuer un certain nombre de modifications et identifie tous leurs effets secondaires;
- met en œuvre toutes les modifications tout en assurant la cohérence des ÉLÉMENTS DE CONFIGURATION du logiciel, y compris le DOSSIER DE GESTION DES RISQUES;
- VÉRIFIE la mise en œuvre des modifications.

Le PROCESSUS de maintenance du logiciel utilise le PROCESSUS de résolution des problèmes de logiciel. Le PROCESSUS de maintenance de logiciel prend les décisions de haut niveau relatives aux rapports de problèmes (s'il existe un problème, s'il a un effet significatif sur la SÉCURITÉ, quelles sont les modifications nécessaires et quand doivent-elles être mises en œuvre) et utilise le PROCESSUS de résolution des problèmes de logiciel pour analyser le RAPPORT DE PROBLÈME afin d'en révéler toutes les implications et de générer d'éventuelles DEMANDES DE MODIFICATION qui identifient tous les ÉLÉMENTS DE CONFIGURATION à modifier, ainsi que toutes les étapes de VÉRIFICATION nécessaires.

B.6.3 Mise en œuvre de la modification

Cette ACTIVITÉ exige que le FABRICANT utilise un PROCESSUS établi pour réaliser la modification. Si un PROCESSUS de maintenance n'a pas été défini, les TÂCHES DE PROCESSUS DE DÉVELOPPEMENT appropriées peuvent être utilisées pour réaliser la modification. Il convient également que le FABRICANT s'assure que la modification n'a pas d'effet négatif sur d'autres parties du LOGICIEL DE DISPOSITIF MÉDICAL. L'analyse de l'effet d'une modification sur l'ensemble du LOGICIEL de DISPOSITIF MÉDICAL est nécessaire sauf si le LOGICIEL de DISPOSITIF MÉDICAL est traité dans le cadre d'un nouveau développement. L'étendue des essais de régression qui seront effectués pour s'assurer que les parties du LOGICIEL de DISPOSITIF MÉDICAL non modifiées se comportent comme avant la modification, doit être justifiée.

because of a PROBLEM REPORT or whether any effects will result from a modification to correct a problem or implement a request. It is especially important to verify through trace or regression analysis that the RISK CONTROL measures built into the device are not adversely changed or modified by the software change that is being implemented as part of the software maintenance ACTIVITY. It is also important to verify that the modified software does not cause a HAZARD or mitigate a RISK in software that previously did not cause a HAZARD or mitigate RISKS. The software safety classification of a SOFTWARE ITEM might have changed if the software modification now can cause a HAZARD or mitigate a RISK.

It is important to distinguish between software maintenance (Clause 6) and software problem resolution (Clause 9).

The focus of the software maintenance PROCESS is an adequate response to feedback arising after release of the SOFTWARE PRODUCT. As part of a MEDICAL DEVICE, the software maintenance PROCESS needs to ensure that:

– SAFETY-related PROBLEM REPORTS are addressed and reported to appropriate regulatory authorities and affected users;

– SOFTWARE PRODUCTS are re-validated and re-released after modification with formal controls that ensure the rectification of the problem and the avoidance of further problems;

– the MANUFACTURER considers what other SOFTWARE PRODUCTS might be affected and takes appropriate action.

The focus of software problem resolution is the operation of a comprehensive control system that:

- analyses PROBLEM REPORTS and identifies all the implications of the problem;
- decides on a number of changes and identifies all their side-effects;
- implements the changes while maintaining the consistency of the software CONFIGURATION ITEMS including the RISK MANAGEMENT FILE;
- VERIFIES the implementation of the changes.

The software maintenance PROCESS uses the software problem resolution PROCESS. The software maintenance PROCESS handles the high-level decisions about the PROBLEM REPORT (whether a problem exists, whether it has a significant effect on SAFETY, what changes are needed and when to implement them), and uses the software problem resolution PROCESS to analyse the PROBLEM REPORT to discover all the implications and to generate possible CHANGE REQUESTS which identify all the CONFIGURATION ITEMS that need to be changed and all the VERIFICATION steps that are necessary.

B.6.3 Modification implementation

This ACTIVITY requires that the MANUFACTURER use an established PROCESS to make the modification. If a maintenance PROCESS has not been defined, the appropriate development PROCESS TASKS can be used to make the modification. The MANUFACTURER should also ensure that the modification does not cause a negative effect on other parts of the MEDICAL DEVICE SOFTWARE. Unless the MEDICAL DEVICE SOFTWARE is treated as a new development, analysis of the effect of a modification on the entire MEDICAL DEVICE SOFTWARE is necessary. A rationale must be made that justifies the amount of REGRESSION TESTING that will be performed to ensure that the portions of the MEDICAL DEVICE SOFTWARE not being modified still perform as they did before the modification was made.

B.7 PROCESSUS de GESTION DES RISQUES du logiciel

La GESTION DES RISQUES DU LOGICIEL fait partie de la GESTION DES RISQUES du DISPOSITIF MÉDICAL dans son ensemble et ne peut être traitée correctement de manière isolée. La présente norme exige l'utilisation d'un PROCESSUS de GESTION DES RISQUES conforme à l'ISO 14971. La GESTION DES RISQUES telle que définie dans l'ISO 14971 traite spécifiquement un cadre de gestion efficace des RISQUES liés à l'utilisation des DISPOSITIFS MÉDICAUX. Une partie de l'ISO 14971 traite de la MAÎTRISE DES RISQUES identifiés, associés à chaque DANGER identifié au cours de l'ANALYSE DE RISQUE. Le PROCESSUS de GESTION DES RISQUES du logiciel défini dans la présente norme fournit des exigences supplémentaires pour LA MAÎTRISE DU RISQUE des logiciels, y compris ceux identifiés au cours de l'ANALYSE DU RISQUE comme contribuant potentiellement à une situation dangereuse ou ceux utilisés pour la MAÎTRISE DES RISQUES des dispositifs médicaux. Le PROCESSUS de GESTION DES RISQUES du logiciel est inclus dans la présente norme pour deux raisons:

a) les destinataires prévus de la présente norme ont besoin de comprendre les exigences minimales des mesures de MAÎTRISE DU RISQUE du logiciel dans leur domaine de compétence;

b) la norme générale de GESTION DES RISQUES l'ISO 14971, citée comme référence normative dans la présente norme, ne traite pas de manière spécifique de la MAÎTRISE DU RISQUE du logiciel ni du positionnement de la MAÎTRISE DU RISQUE dans le cycle de vie de développement du logiciel.

La GESTION DES RISQUES du logiciel fait partie de la GESTION DES RISQUES du DISPOSITIF MÉDICAL dans son ensemble. Les plans, les procédures et la documentation requis pour les ACTIVITÉS de GESTION DES RISQUES du logiciel peuvent être une série de documents séparés ou un seul document ou peuvent encore être intégrés aux ACTIVITÉS et à la documentation de GESTION DES RISQUES du DISPOSITIF MÉDICAL tant que toutes les exigences de la présente norme sont remplies.

B.7.1 Analyse du logiciel en termes de contribution à des situations dangereuses

Il est prévu que l'analyse des PHÉNOMÈNES DANGEREUX du dispositif identifie les situations dangereuses ainsi que les mesures de MAÎTRISE DU RISQUE correspondantes afin d'en réduire la probabilité et/ou la gravité à un niveau acceptable. Il est également prévu que des mesures de MAÎTRISE DU RISQUE seront affectées à des fonctions logicielles conçues pour mettre en œuvre ces mesures.

Cependant, il n'est pas prévu que toutes les situations dangereuses du dispositif puissent être identifiées avant que l'ARCHITECTURE du logiciel ne soit produite. A cette étape, on sait la manière dont les fonctions logicielles seront mises en œuvre dans les composants logiciels et la faisabilité des mesures de MAÎTRISE DU RISQUE attribuées aux fonctions logicielles peut être ÉVALUÉE. Il convient à ce moment là de réviser l'analyse des PHÉNOMÈNES DANGEREUX des dispositifs afin d'inclure:

- les situations dangereuses révisées;
- les mesures de MAÎTRISE DU RISQUE et les exigences logicielles révisées;
- les nouvelles situations dangereuses résultant du logiciel, par exemple des situations dangereuses liées à des facteurs humains.

Il convient que l'ARCHITECTURE de logiciel inclut des stratégies crédibles de découpage des composants logiciels de façon à ce qu'il n'y ait pas d'interaction contraire à la SÉCURITÉ.

B.7 Software RISK MANAGEMENT PROCESS

Software RISK MANAGEMENT is a part of overall MEDICAL DEVICE RISK MANAGEMENT and cannot be adequately addressed in isolation. This standard requires the use of a RISK MANAGEMENT PROCESS that is compliant with ISO 14971. RISK MANAGEMENT as defined in ISO 14971 deals specifically with a framework for effective management of the RISKS associated with the use of MEDICAL DEVICES. One portion of ISO 14971 pertains to control of identified RISKS associated with each HAZARD identified during the RISK ANALYSIS. The software RISK MANAGEMENT PROCESS in this standard is intended to provide additional requirements for RISK CONTROL for software, including software that has been identified during the RISK ANALYSIS as potentially contributing to a hazardous situation, or software that is used to control MEDICAL DEVICE RISKS. The software RISK MANAGEMENT PROCESS is included in this standard for two reasons.

a) the intended audience of this standard needs to understand minimum requirements for RISK CONTROL measures in their area of responsibility—software;
b) the general RISK MANAGEMENT standard, ISO 14971, provided as a normative reference in this standard, does not specifically address the RISK CONTROL of software and the placement of RISK CONTROL in the software development life cycle.

Software RISK MANAGEMENT is a part of overall MEDICAL DEVICE RISK MANAGEMENT. Plans, procedures, and documentation required for the software RISK MANAGEMENT ACTIVITIES can be a series of separate documents or a single document, or they can be integrated with the MEDICAL DEVICE RISK MANAGEMENT ACTIVITIES and documentation as long as all requirements in this standard are met.

B.7.1 Analysis of software contributing to hazardous situations

It is expected that the device HAZARD analysis will identify hazardous situations and corresponding RISK CONTROL measures to reduce the probability and/or severity of those hazardous situations to an acceptable level. It is also expected that the RISK CONTROL measures will be assigned to software functions that are expected to implement those RISK CONTROL measures.

However, it is not expected that all device hazardous situations can be identified until the software ARCHITECTURE has been produced. At that time it is known how software functions will be implemented in software components, and the practicality of the RISK CONTROL measures assigned to software functions can be EVALUATED. At that time the device HAZARD analysis should be revised to include:

- revised hazardous situations;
- revised RISK CONTROL measures and software requirements;
- new hazardous situations arising from software, for example hazardous situations related to human factors.

The software ARCHITECTURE should include credible strategies for segregating software components so that they do not interact in unsafe ways.

B.8 PROCESSUS de gestion de configuration du logiciel

Le PROCESSUS de gestion de la configuration du logiciel est un PROCESSUS appliquant des procédures administratives et techniques pendant le cycle de vie du logiciel afin d'identifier et de définir des ÉLÉMENTS LOGICIELS, y compris la documentation, dans un SYSTÈME DONNÉ; de maîtriser les modifications et les versions des ÉLÉMENTS LOGICIELS; et de consigner et de rendre compte de l'état des ÉLÉMENTS LOGICIELS et des DEMANDES DE MODIFICATION. La GESTION DE LA CONFIGURATION DU LOGICIEL est nécessaire pour recréer un ÉLÉMENT LOGICIEL, en identifier ses parties constitutives et fournir un historique des modifications qu'il a subies.

B.8.1 Identification de la configuration

Cette ACTIVITÉ exige que le FABRICANT identifie de manière univoque des éléments de CONFIGURATION DU LOGICIEL et LEURS VERSIONS. Ceci est nécessaire à l'identification des ÉLÉMENTS DE CONFIGURATION du logiciel et des versions incluses dans le LOGICIEL DE DISPOSITIF MÉDICAL.

B.8.2 Maîtrise des modifications

Cette ACTIVITÉ exige que le FABRICANT maîtrise les modifications apportées aux ÉLÉMENTS DE CONFIGURATION du logiciel et qu'il consigne les informations identifiant les DEMANDES DE MODIFICATION et fournissant la documentation relative à leur prise en charge. Cette ACTIVITÉ est nécessaire pour s'assurer que des modifications non autorisées ou involontaires ne sont pas effectuées sur des éléments de configuration du LOGICIEL et que les DEMANDES DE MODIFICATION approuvées sont pleinement mises en œuvre et vérifiées.

Les DEMANDES DE MODIFICATION peuvent être approuvées par un comité de MAÎTRISE DES MODIFICATIONS ou par un directeur ou un responsable technique selon le plan de gestion de la configuration du logiciel. La TRAÇABILITÉ des DEMANDES DE MODIFICATION approuvées est rattachée à la modification réelle et à la VÉRIFICATION du logiciel. L'exigence est que chaque modification réelle soit reliée à une DEMANDE DE MODIFICATION et qu'il existe une documentation montrant que la DEMANDE DE MODIFICATION a été approuvée. La documentation pourrait être les procès-verbaux de réunion du comité de maîtrise des modifications, une signature d'approbation ou un enregistrement dans une base de données.

B.8.3 Documentation relative à l'état de la configuration

Cette ACTIVITÉ exige que le FABRICANT conserve des enregistrements de l'historique des éléments de CONFIGURATION DU LOGICIEL. Cette ACTIVITÉ est nécessaire pour déterminer quand et pourquoi des modifications ont été effectuées. Un accès à ces informations est nécessaire pour s'assurer que les ÉLÉMENTS DE CONFIGURATION du logiciel comportent uniquement des modifications autorisées.

B.9 PROCESSUS de résolution de problème logiciel

Le PROCESSUS de résolution des problèmes de logiciel est un PROCESSUS d'analyse et de résolution des problèmes (y compris les non-conformités), quelle que soit leur nature ou source, y compris ceux découverts pendant le développement, la maintenance ou autres PROCESSUS. L'objectif est de fournir en temps opportun un moyen responsable et documenté pour s'assurer que les problèmes décelés sont analysés et résolus et que les tendances sont reconnues. Ce PROCESSUS est quelquefois appelé «localisation des défauts» dans la littérature d'ingénierie logicielle. Il est appelé «résolution des problèmes» dans les normes ISO/CEI 12207 [9] et CEI 60601-1-4 [2], Amendement 1. Dans la présente norme, nous avons choisi de l'appeler «résolution des problèmes de logiciel».

B.8 Software configuration management PROCESS

The software configuration management PROCESS is a PROCESS of applying administrative and technical procedures throughout the software life cycle to identify and define SOFTWARE ITEMS, including documentation, in a SYSTEM; control modifications and releases of the items; and document and report the status of the items and CHANGE REQUESTS. Software configuration management is necessary to recreate a SOFTWARE ITEM, to identify its constituent parts, and to provide a history of the changes that have been made to it.

B.8.1 Configuration identification

This ACTIVITY requires the MANUFACTURER to uniquely identify software CONFIGURATION ITEMS and their VERSIONS. This identification is necessary to identify the software CONFIGURATION ITEMS and the VERSIONS that are included in the MEDICAL DEVICE SOFTWARE.

B.8.2 Change control

This ACTIVITY requires the MANUFACTURER to control changes of the software CONFIGURATION ITEMS and to document information identifying CHANGE REQUESTS and providing documentation about their disposition. This ACTIVITY is necessary to ensure that unauthorized or unintended changes are not made to the software CONFIGURATION ITEMS and to ensure that approved CHANGE REQUESTS are implemented fully and verified.

CHANGE REQUESTS can be approved by a change control board or by a manager or technical lead according to the software configuration management plan. Approved CHANGE REQUESTS are made traceable to the actual modification and VERIFICATION of the software. The requirement is that each actual change be linked to a CHANGE REQUEST and that documentation exists to show that the CHANGE REQUEST was approved. The documentation might be change control board minutes, an approval signature, or a record in a database.

B.8.3 Configuration status accounting

This ACTIVITY requires the MANUFACTURER to maintain records of the history of the software CONFIGURATION ITEMS. This ACTIVITY is necessary to determine when and why changes were made. Access to this information is necessary to ensure that software CONFIGURATION ITEMS contain only authorized modifications.

B.9 Software problem resolution PROCESS

The software problem resolution PROCESS is a PROCESS for analyzing and resolving the problems (including non-conformances), whatever their nature or source, including those discovered during the execution of development, maintenance, or other PROCESSES. The objective is to provide a timely, responsible, and documented means to ensure that discovered problems are analyzed and resolved and that trends are recognized. This PROCESS is sometimes called “defect tracking” in software engineering literature. It is called “problem resolution” in ISO/IEC 12207 [9] and IEC 60601-1-4 [2], Amendment 1. We have chosen to call it “software problem resolution” in this standard.

Cette ACTIVITÉ exige que le FABRICANT utilise le PROCESSUS de résolution des problèmes de logiciel lorsqu'un problème ou une non-conformité est identifié. Cette ACTIVITÉ est nécessaire pour s'assurer que les problèmes décelés sont analysés et évalués en termes de pertinence éventuelle vis-à-vis de la SÉCURITÉ (comme spécifié dans l'ISO 14971).

La manière de traiter les problèmes ou les non-conformités fait l'objet de plan(s) ou de procédures de développement du logiciel, comme exigé en 5.1. Ceci implique qu'il faut spécifier à chaque stade du cycle de vie les aspects du PROCESSUS de résolution des problèmes de logiciel qui seront formels et documentés ainsi que le moment où les problèmes et non-conformités doivent être introduits dans le PROCESSUS de résolution des problèmes de logiciel.

This ACTIVITY requires that the MANUFACTURER use the software problem resolution PROCESS when a problem or non-conformance is identified. This ACTIVITY is necessary to ensure that discovered problems are analyzed and EVALUATED for possible relevance to SAFETY (as specified in ISO 14971).

Software development plan(s) or procedures, as required in 5.1, are to address how problems or non-conformances will be handled. This includes specifying at each stage of the life cycle the aspects of the software problem resolution PROCESS that will be formal and documented as well as when problems and nonconformities are to be entered into the software problem resolution PROCESS.

Annexe C
(informative)

Relations avec d'autres normes

C.1 Généralités

La présente norme s'applique au développement et à la maintenance des logiciels de DISPOSITIFS MÉDICAUX. Le logiciel est considéré être un sous-système du DISPOSITIF MÉDICAL ou est lui-même un DISPOSITIF MÉDICAL. La présente norme doit être utilisée conjointement à d'autres normes pertinentes pour le développement d'un DISPOSITIF MÉDICAL.

Les normes de gestion de DISPOSITIFS MÉDICAUX telles que l'ISO 13485 [7] (voir l'Article C.2 et l'Annexe D) et l'ISO 14971 (voir l'Annexe C.3) fournissent un environnement de gestion qui établit une base permettant à une organisation de développer des produits. Les normes de sécurité telles que la CEI 60601-1 [1] (voir l'Article C.4) et la CEI 61010-1 [4] (voir l'Article C.5) donnent des instructions spécifiques pour la création de DISPOSITIFS MÉDICAUX sûrs. Lorsque le logiciel fait partie intégrante de ces DISPOSITIFS MÉDICAUX, la CEI 62304 fournit des instructions plus détaillées quant aux exigences de développement et de maintenance de logiciels DE DISPOSITIFS MÉDICAUX sûrs. De nombreuses autres normes telles que l'ISO/CEI 12207 [9] (voir l'Article C.6), la CEI 61508-3 [3] (voir l'Article C.7), et l'ISO/CEI 90003 [11] peuvent être consultées en tant que sources de méthodes, d'outils et de techniques qui peuvent être utilisés pour mettre en œuvre les exigences de la CEI 62304. La Figure C.1 illustre les liens entre ces normes.

Lorsque les articles ou les exigences d'autres normes sont cités, les termes définis dans les éléments cités sont des termes définis dans une norme autre que la présente norme.

Annex C
(informative)

Relationship to other standards

C.1 General

This standard applies to the development and maintenance of MEDICAL DEVICE SOFTWARE. The software is considered a subsystem of the MEDICAL DEVICE or is itself a MEDICAL DEVICE. This standard is to be used together with other appropriate standards when developing a MEDICAL DEVICE.

MEDICAL DEVICE management standards such as ISO 13485 [7] (see C.2 and Annex D) and ISO 14971 (see Annex 0) provide a management environment that lays a foundation for an organization to develop products. Safety standards such as IEC 60601-1 [1] (see Annex C.4) and IEC 61010-1 [4] (see Annex C.5) give specific direction for creating safe MEDICAL DEVICES. When software is a part of these MEDICAL DEVICES, IEC 62304 provides more detailed direction on what is required to develop and maintain safe MEDICAL DEVICE SOFTWARE. Many other standards such as ISO/IEC 12207 [9] (see Annex C.6), IEC 61508-3 [3] (see Annex C.7) and ISO/IEC 90003 [11] can be looked to as a source of methods, tools and techniques that can be used to implement the requirements in IEC 62304. Figure C.1 shows the relationship of these standards.

Where clauses or requirements from other standards are quoted, defined terms in the quoted items are terms that are defined in the other standard, not defined terms in this standard.

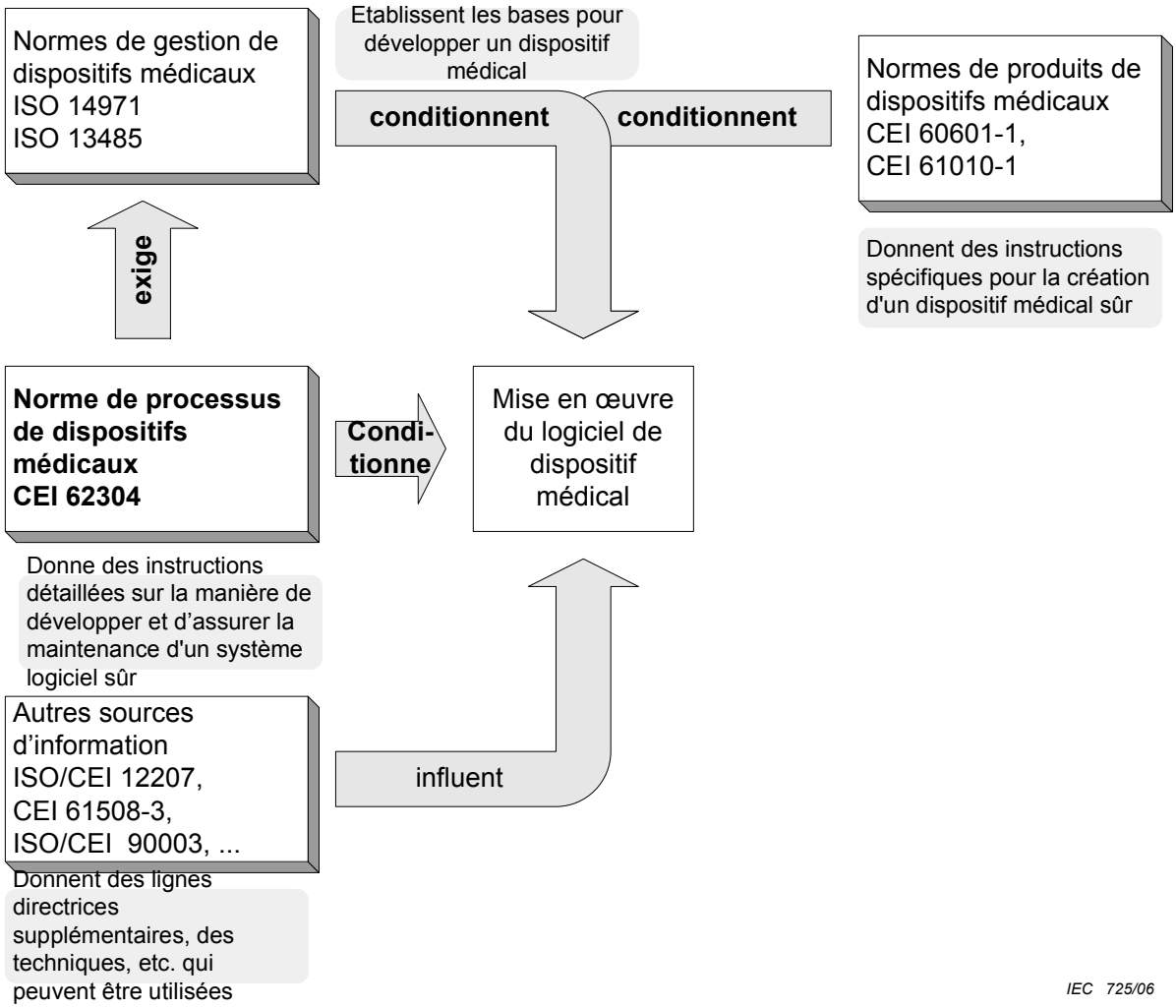

Figure C.1 – Relation des principales normes de DISPOSITIFS MÉDICAUX avec la CEI 62304

C.2 Relation avec l'ISO 13485

La présente norme exige que le FABRICANT utilise un système de management de la qualité. Lorsqu'un FABRICANT utilise l'ISO 13485 [7], les exigences de l'ISO 62304 font directement référence à certaines des exigences de l'ISO 13485 comme illustré dans le Tableau C.1.

Tableau C.1 – Relation avec l'ISO 13485:2003

Article de la CEI 62304	Paragraphe correspondant de l'ISO 13485:2003
5.1 Planification du développement du logiciel	7.3.1 Planification de la conception et du développement
5.2 Analyses des exigences du logiciel	7.3.2 Eléments d'entrée de la conception et du développement
5.3 Conception architecturale du logiciel	
5.4 Conception détaillée du logiciel	
5.5 Mise en œuvre et vérification des UNITÉS LOGICIELLES	
5.6 Intégration et essai d'intégration du logiciel	
5.7 Essais du SYSTÈME LOGICIEL	7.3.3 Résultats de sortie de la conception et du développement 7.3.4 Revue de la conception et du développement
5.8 Diffusion du logiciel	7.3.5 Vérification de la conception et du développement 7.3.6 Validation de la conception et du développement

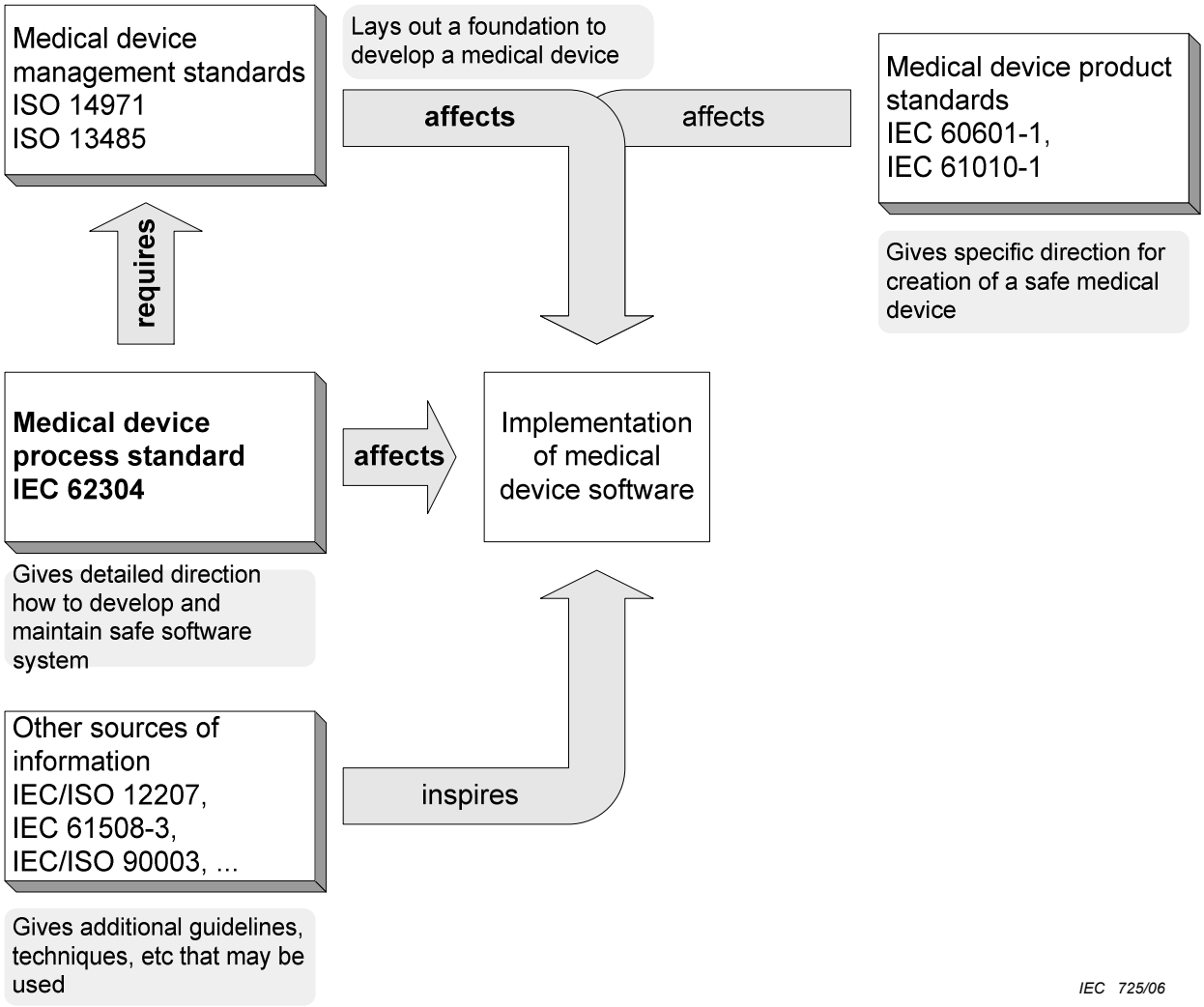

Figure C.1 – Relationship of key MEDICAL DEVICE standards to IEC 62304

C.2 Relationship to ISO 13485

This standard requires that the MANUFACTURER employs a quality management system. When a MANUFACTURER uses ISO 13485 [7], the requirements of ISO 62304 directly relate to some of the requirements of ISO 13485 as shown in Table C.1.

Table C.1 – Relationship to ISO 13485:2003

IEC 62304 clause	Related clause of ISO 13485:2003
5.1 Software development planning	7.3.1 Design and development planning
5.2 Software requirements analysis	7.3.2 Design and development inputs
5.3 Software ARCHITECTURAL design	
5.4 Software detailed design	
5.5 SOFTWARE UNIT implementation and verification	
5.6 Software integration and integration testing	
5.7 SOFTWARE SYSTEM testing	7.3.3 Design and development outputs 7.3.4 Design and development review
5.8 Software release	7.3.5 Design and development verification 7.3.6 Design and development validation

Tableau C.1 *(suite)*

Article de la CEI 62304	Paragraphe correspondant de l'ISO 13485:2003
6.1 Etablissement du plan de maintenance du logiciel	7.3.7 Maîtrise des modifications de la conception et du développement
6.2 Analyse des problèmes et des modifications	
6.3 Mise en œuvre de la modification	7.3.5 Vérification de la conception et du développement 7.3.6 Validation de la conception et du développement
7.1 Analyse du logiciel en termes de contribution à des situations dangereuses	
7.2 MESURES DE maîtrise DU RISQUE	
7.3 VÉRIFICATION des MESURES DE MAÎTRISE DU RISQUE	
7.4 GESTION DES RISQUES des modifications du logiciel	
8.1 Identification de la configuration	7.5.3 Identification et TRAÇABILITÉ
8.2 Maîtrise des modifications	7.5.3 Identification et TRAÇABILITÉ
8.3 Documentation relative à l'état de la configuration	
9 PROCESSUS de résolution de problème logiciel	

C.3 Relation avec l'ISO 14971

Le Tableau C.2 indique les articles dans lesquels la CEI 62304 détaille les exigences du PROCESSUS de GESTION DES RISQUES exigé par l'ISO 14971.

Tableau C.2 – Relation avec l'ISO 14971:2000

Article de l'IS0 14971:2000	Article correspondant de la CEI 62304
4.1 Procédure d'ANALYSE DU RISQUE	
4.2 Usage/objet prévu et identification des caractéristiques relatives à la SÉCURITÉ du DISPOSITIF MÉDICAL	
4.3 Identification des PHÉNOMÈNES DANGEREUX connus ou prévisibles	7.1 Analyse du logiciel en termes de contribution à des situations dangereuses
4.4 Estimation du (des) RISQUE(S) pour chaque phénomène dangereux	4.3 Classification de sécurité du logiciel
5 Évaluation du RISQUE	
6.1 Réduction du RISQUE	
6.2 Analyse des options	7.2.1 Définition DES MESURES DE MAÎTRISE DU RISQUE
6.3 Mise en œuvre des mesures de MAÎTRISE DU RISQUE	7.2.2 Mesures de MAÎTRISE DU RISQUE mises en œuvre dans le logiciel 7.3.1 Vérification des mesures de MAÎTRISE DU RISQUE
6.4 Évaluation du RISQUE résiduel	
6.5 ANALYSE RISQUE/bénéfice	
6.6 Autres PHÉNOMÈNES DANGEREUX engendrés	7.3.2 Consignation de toutes nouvelles séquences d'événements
6.7 Complétude de l'évaluation du RISQUE	
7 Évaluation du RISQUE résiduel global	
8 RAPPORT DE GESTION DES RISQUES	7.3.3 Consignation de la TRAÇABILITÉ
9 Information post-production	7.4 GESTION DES RISQUES des modifications du logiciel

Table C.1 *(continued)*

IEC 62304 clause	Related clause of ISO 13485:2003
6.1 Establish software maintenance plan	7.3.7 Control of design and development changes
6.2 Problem and modification analysis	
6.3 Modification implementation	7.3.5 Design and development verification 7.3.6 Design and development validation
7.1 Analysis of software contributing to hazardous situations	
7.2 RISK CONTROL measures	
7.3 VERIFICATION of RISK CONTROL measures	
7.4 RISK MANAGEMENT of software changes	
8.1 Configuration identification	7.5.3 Identification and TRACEABILITY
8.2 Change control	7.5.3 Identification and TRACEABILITY
8.3 Configuration status accounting	
9 Software problem resolution PROCESS	

C.3 Relationship to ISO 14971

Table C.2 shows the areas where IEC 62304 amplifies requirements for the RISK MANAGEMENT PROCESS required by ISO 14971.

Table C.2 – Relationship to ISO 14971:2000

ISO 14971:2000 clause	Related clause of IEC 62304
4.1 RISK ANALYSIS procedure	
4.2 Intended use/intended purpose and identification of characteristics related to the SAFETY of the MEDICAL DEVICE	
4.3 Identification of known or foreseeable HAZARDS	7.1 Analysis of software contributing to hazardous situations
4.4 Estimation of the RISK(S) for each HAZARD	4.3 Software safety classification
5 RISK evaluation	
6.1 RISK reduction	
6.2 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 7.3.1 Verify RISK CONTROL measures
6.4 Residual RISK evaluation	
6.5 RISK/benefit analysis	
6.6 Other generated HAZARDS	7.3.2 Document any new sequences of events
6.7 Completeness of RISK evaluation	
7 Overall residual RISK evaluation	
8 RISK MANAGEMENT report	7.3.3 Document TRACEABILITY
9 Post-production information	7.4 RISK MANAGEMENT of software changes

C.4 Relation avec les exigences de SEMP de la CEI 60601-1:2005

C.4.1 Généralités

Les exigences applicables au logiciel sont un sous-ensemble des exigences applicables à un système électromédical programmable (SEMP). La présente norme identifie les exigences de logiciel qui viennent en supplément aux exigences de la CEI 60601-1 [1] pour les SEMP mais qui ne sont pas incompatibles avec ces exigences. Etant donné que les systèmes SEMP incluent des éléments qui ne sont pas logiciels, toutes les exigences de la CEI 60601-1 pour les SEMP ne sont pas traitées dans la présente norme.

C.4.2 Relation du logiciel avec le développement du système SEMP

En utilisant le modèle en V illustré à la Figure C.2 pour décrire les différents événements du développement d'un SEMP, on peut voir que les exigences de la présente norme de logiciel s'appliquent au niveau composant de SEMP, depuis la spécification des exigences du logiciel jusqu'à l'intégration des ÉLÉMENTS LOGICIELS dans un SYSTÈME LOGICIEL. Le SYSTÈME LOGICIEL fait partie d'un sous-système électrique programmable (SSEP), qui est une partie d'un SEMP.

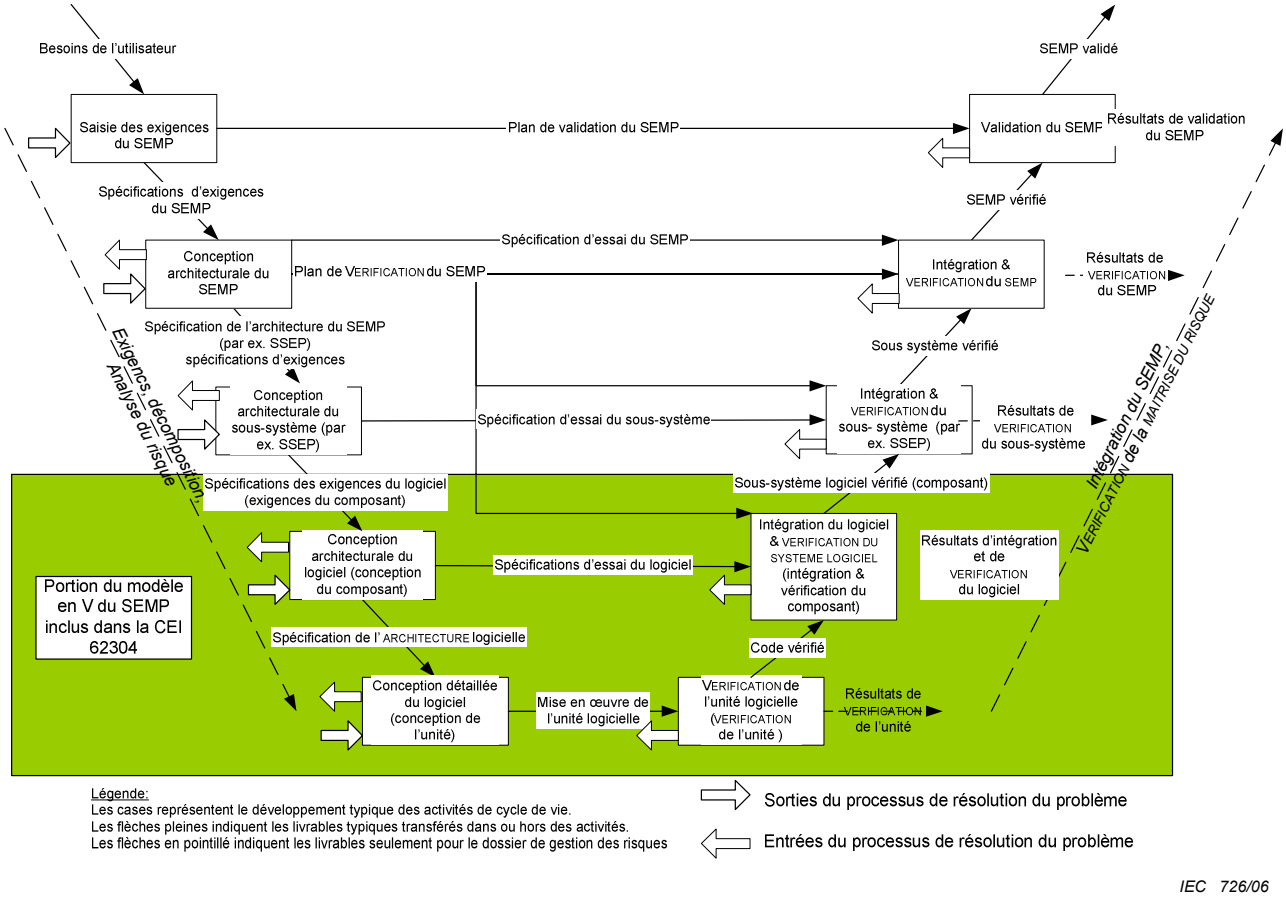

Figure C.2 – Logiciel comme partie du modèle en V

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

C.4.1 General

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 [1] for PEMS. Because PEMS include elements that are not software, not all of the requirements of IEC 60601-1 for PEMS are addressed in this standard.

C.4.2 Software relationship to PEMS development

By using the V-model illustrated in Figure C.2 to describe what occurs during a PEMS development, it can be seen that the requirements of this software standard apply at the PEMS component level, from the specification of the software requirements to the integration of the SOFTWARE ITEMS into a SOFTWARE SYSTEM. This SOFTWARE SYSTEM is a part of a programmable electrical subsystem (PESS), which is a part of a PEMS.

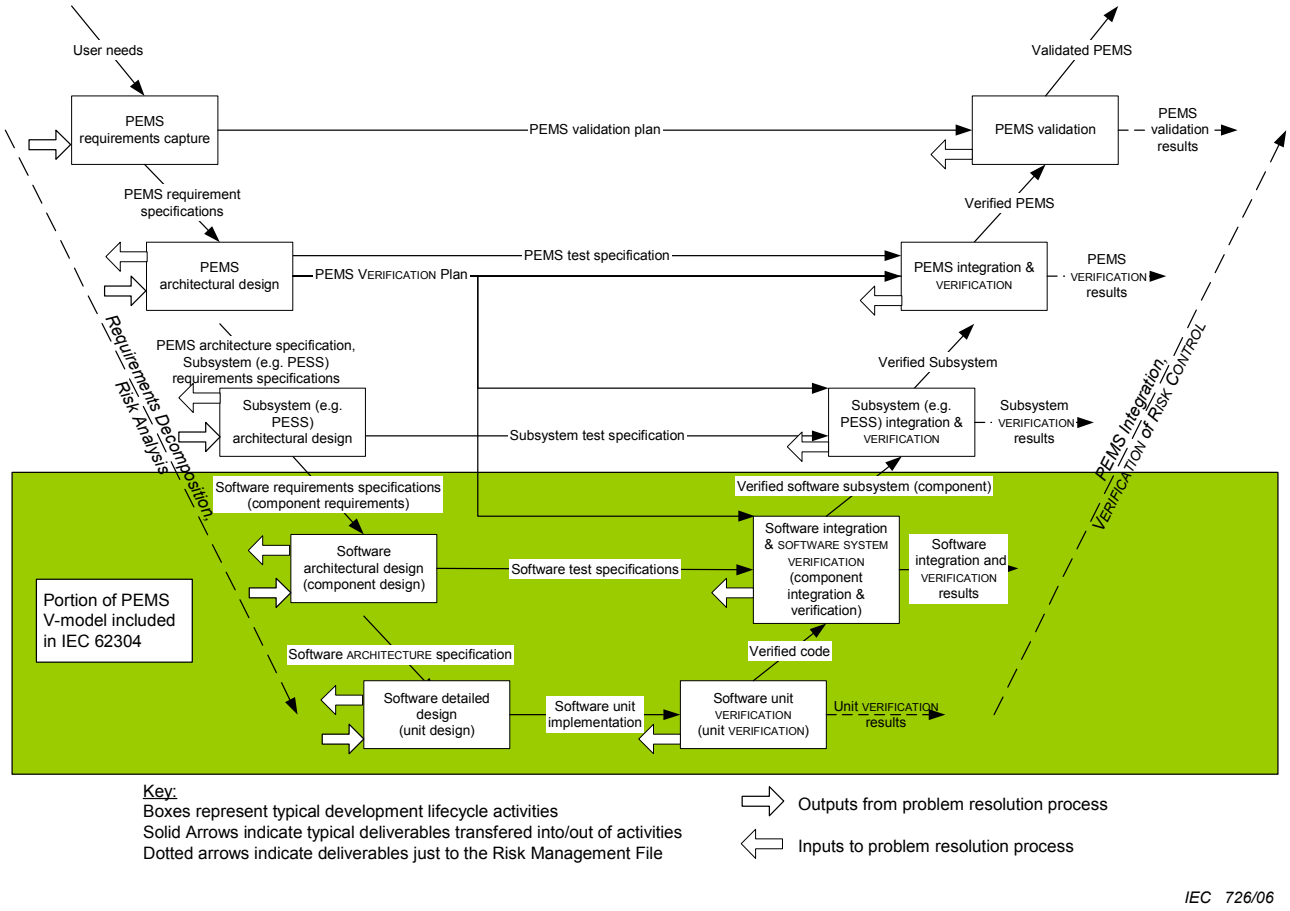

Figure C.2 – Software as part of the V-model

C.4.3 PROCESSUS de développement

La conformité au PROCESSUS de développement du logiciel de la présente norme (Article 5) exige qu'un plan de développement du logiciel soit spécifié et suivi; elle n'exige pas l'utilisation d'un modèle particulier de cycle de vie mais elle exige que le plan comprenne certaines ACTIVITÉs et ait certains attributs. Ces exigences sont liées aux exigences du SEMP de la CEI 60601-1 pour le cycle de vie de développement, la spécification des exigences, L'ARCHITECTURE, la conception et la mise en œuvre et la VÉRIFICATION. Les exigences de la présente norme donnent à propos du développement du logiciel des informations plus détaillées que celles contenues dans la CEI 60601-1.

C.4.4 PROCESSUS de maintenance

La conformité au PROCESSUS de maintenance du logiciel de la présente norme (Article 6) exige que des procédures soient établies et suivies lorsque des modifications sont apportées au logiciel. Ces exigences correspondent à l'exigence de modification d'un SEMP de la CEI 60601-1. Les exigences de la présente norme fournissent, en ce qui concerne ce qui doit être effectué pour la maintenance du logiciel, des informations plus détaillées que celles des exigences pour la modification du SEMP dans la CEI 60601-1.

C.4.5 Autres PROCESSUS

Les autres PROCESSUS de la présente norme spécifient des exigences supplémentaires pour le logiciel au-delà des exigences similaires pour le SEMP dans la CEI 60601-1. Dans la plupart des cas, la CEI 60601-1 fournit une exigence d'ordre général pour le SEMP, tandis que la présente norme approfondit les PROCESSUS.

Le PROCESSUS de GESTION DES RISQUES du logiciel dans la présente norme correspond aux exigences supplémentaires de GESTION DES RISQUES identifiées pour le SEMP dans la CEI 60601-1.

Le PROCESSUS de résolution des problèmes de logiciel dans la présente norme correspond aux exigences de résolution des problèmes pour le SEMP dans la CEI 60601-1.

Le PROCESSUS de gestion de la configuration du logiciel dans la présente norme spécifie des exigences supplémentaires qui n'existent pas pour le SEMP dans la CEI 60601-1 sauf pour ce qui concerne la documentation.

C.4.6 Traitement des exigences du SEMP dans la CEI 60601-1

Le Tableau C.3 illustre les exigences SEMP de la CEI 60601-1 et les exigences correspondantes de la présente norme.

C.4.3 Development PROCESS

Compliance with the software development PROCESS of this standard (Clause 5) requires that a software development plan be specified and then followed; it does not require that any particular life cycle model is used, but it does require that the plan include certain ACTIVITIES and have certain attributes. These requirements relate to the PEMS requirements in IEC 60601-1 for development life cycle, requirement specification, ARCHITECTURE, design and implementation, and VERIFICATION. The requirements in this standard provide greater detail about software development than those in IEC 60601-1.

C.4.4 Maintenance PROCESS

Compliance with the software maintenance PROCESS of this standard (Clause 6) requires that procedures be established and followed when changes to software are made. These requirements correspond to the requirement in IEC 60601-1 for modification of a PEMS. The requirements in this standard for software maintenance provide greater detail about what must be done for software maintenance than the requirements for PEMS modification in IEC 60601-1.

C.4.5 Other PROCESSES

The other PROCESSES in this standard specify additional requirements for software beyond the similar requirements for PEMS in IEC 60601-1. In most cases, there is a general requirement for PEMS in IEC 60601-1, which the PROCESSES in this standard expand upon.

The software RISK MANAGEMENT PROCESS in this standard corresponds to the additional RISK MANAGEMENT requirements identified for PEMS in IEC 60601-1.

The software problem resolution PROCESS in this standard corresponds to the problem resolution requirement for PEMS in IEC 60601-1.

The software configuration management PROCESS in this standard specifies additional requirements that are not present for PEMS in IEC 60601-1 except for documentation.

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

Table C.3 shows the PEMS requirements of IEC 60601-1 and the corresponding requirements in this standard.

Tableau C.3 – Relation avec la CEI 60601-1

Exigences pour un SEMP de la CEI 60601-1:2005	Exigences de la CEI 62304 relatives au sous-système logiciel d'un SEMP
14.1 Généralités Les exigences de ce paragraphe doivent s'appliquer au SEMP sauf si: – le SSEP ne garantit pas la SÉCURITÉ DE BASE ni les PERFORMANCES ESSENTIELLES;ou – l'application de l'ISO 14971 démontre que la défaillance du SSEP n'entraîne pas un RISQUE inacceptable.	**4.3** Classification de sécurité du logiciel Les exigences de la CEI 60601-1 pour le système SEMP s'appliqueraient uniquement aux classes de SÉCURITÉ du logiciel B et C. La présente norme inclut certaines exigences pour la classe de SÉCURITÉ de logiciel A.
14.2 Documentation En supplément aux enregistrements et documents exigés par l'ISO 14971, les documents produits par application de l'Article 14 doivent être maintenus et faire partie dU DOSSIER DE GESTION DES RISQUES.	**4.2 GESTION DES** RISQUES
Les documents exigés par l'Article 14 doivent être revus, approuvés, édités et modifiés conformément à une procédure formelle de maîtrise des documents.	**5.1 Planification du développement du logiciel** Outre les exigences spécifiques à l'ACTIVITÉ de planification du développement du logiciel, il est exigé par l'ISO que soient conservés des documents qui font partie du DOSSIER DE GESTION DES RISQUES. En outre, pour les documents qui sont exigés par le système qualité, l'ISO 13485 [7]exige la maîtrise des documents.
14.3 PLAN DE GESTION DES RISQUES Le plan de gestion des RISQUES exigé au 3.5 de l'ISO 14971, doit également inclure une référence au plan de validation du SEMP (voir 14.11).	N'est pas spécifiquement exigé. Il n'y a pas de plan spécifique de validation du logiciel. Le plan de validation du SEMP se situe au niveau du système et ainsi il est hors du domaine d'application de la présente norme de logiciel. La présente norme exige une TRAÇABILITÉ depuis le PHÉNOMÈNE DANGEREUX en passant par la cause spécifique au logiciel et les mesures de MAÎTRISE DU RISQUE jusqu'à la VÉRIFICATION de la mesure de MAÎTRISE DU RISQUE (voir 7.3)
14.4 Cycle de développement du système SEMP Le cycle de développement d'un SEMP doit être documenté.	**5.1 Planification du développement du logiciel** 5.1.1 Plan de développement du logiciel Les éléments couverts par le plan de développement du logiciel constituent un cycle de développement du logiciel.
Le cycle de développement d'un SEMP doit comporter un ensemble de jalons bien défini.	
A chaque jalon, les ACTIVITÉS qui doivent être menées à bien et les méthodes de vérification à appliquer à ces ACTIVITÉS, doivent être définies.	5.1.6 Planning de VÉRIFICATION du logiciel Les TÂCHES DE VÉRIFICATION, les critères de réception et d'étapes doivent être planifiés
Chaque ACTIVITÉ doit être définie en indiquant ses éléments d'entrée et de sortie.	5.1.1 Plan de développement du logiciel Les ACTIVITÉS sont définies dans la présente norme. La documentation à produire est définie dans chaque ACTIVITÉ.
Chaque jalon doit identifier les ACTIVITÉs de gestion des RISQUES qui doivent être menées à bien avant ce jalon.	
Le cycle de développement d'un SEMP doit être adapté à chaque développement spécifique en élaborant des plans qui détaillent les ACTIVITÉs, des jalons et des plannings.	5.1.1 Plan de développement du logiciel La présente norme permet de documenter le cycle de vie de développement dans le plan de développement. Ce qui signifie que le plan de développement contient un cycle de vie de développement adapté.
Le cycle de développement d'un SEMP doit inclure les exigences en matière de documentation.	5.1.1 Plan de développement du logiciel 5.1.8 Planification de la documentation
14.5 Résolution des problèmes Le cas échéant, un système documenté de résolution des problèmes pendant et entre toutes les phases et ACTIVITÉS du cycle de développement d'un système SEMP doit être développé et maintenu.	**9 Processus de résolution de** problème logiciel

Table C.3 – Relationship to IEC 60601-1

PEMS requirements from IEC 60601-1:2005	Requirements of IEC 62304 relating to the software subsystem of a PEMS
14.1 General The requirements of this clause shall apply to PEMS unless: – the PESS provides no BASIC SAFETY or ESSENTIAL PERFORMANCE; or – the application of ISO 14971 demonstrates that the failure of the PESS does not lead to an unacceptable RISK.	**4.3 Software safety classification** 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.
14.2 Documentation In addition to the records and documents required by ISO 14971, the documents produced from application of Clause 14 shall be maintained and shall form part of the RISK MANAGEMENT FILE.	**4.2 RISK MANAGEMENT**
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** 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 [7] requires control of the documents.
14.3 RISK MANAGEMENT PLAN The RISK MANAGEMENT plan required by 3.5 of ISO 14971 shall also include a reference to the PEMS VALIDATION plan (see 14.11).	Not specifically required. 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 A PEMS DEVELOPMENT LIFE-CYCLE shall be documented.	**5.1 Software development planning** 5.1.1 Software development plan 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 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 ACTIVITIES are defined in this standard. Documentation to be produced is defined in each ACTIVITY.
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 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 5.1.8 Documentation planning
14.5 Problem resolution 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**

Tableau C.3 *(suite)*

Exigences pour un SEMP de la CEI 60601-1:2005	Exigences de la CEI 62304 relatives au sous-système logiciel d'un SEMP
En fonction du type de produit, le système de résolution des problèmes peut: – être consigné par écrit en tant que partie du cycle de développement du SEMP; – permettre de rendre compte de problèmes potentiels ou existants affectant la SÉCURITÉ DE BASE OU les PERFORMANCES ESSENTIELLES – inclure une évaluation de chaque problème pour les RISQUES associés; – identifier les critères à satisfaire pour prononcer une conclusion; – identifier les actions à entreprendre pour résoudre chaque problème.	5.1.1 Plan de développement du logiciel 9.1 Elaboration des RAPPORTS DE PROBLÈME
14.6 PROCESSUS de GESTION DES RISQUES	**7 Processus DE GESTION DES** RISQUES DU LOGICIEL
14.6.1 Identification des PHÉNOMÈNES DANGEREUX connus ou prévisibles Lors de l'élaboration de la liste des PHÉNOMÈNES DANGEREUX connus ou prévisibles, le FABRICANT doit tenir compte des PHÉNOMÈNES DANGEREUX liés aux aspects logiciels et matériels du système SEMP y compris ceux relatifs aux liaisons réseau/données, composants en provenance de tierce partie et sous-systèmes hérités.	7.1 Analyse du logiciel en termes de contribution à des situations dangereuses La présente norme ne mentionne pas spécifiquement les liaisons réseau/données
14.6.2 MAÎTRISE DU RISQUE Des outils et des procédures correctement validés doivent être sélectionnés et identifiés pour la mise en œuvre de chaque mesure de MAÎTRISE DU RISQUE. Ces outils et procédures doivent convenir pour s'assurer que chaque mesure de MAÎTRISE DU RISQUE réduit de manière satisfaisante le(s) risque(s) identifié(s).	5.1.4 Planification des normes, méthodes et outils de développement du logiciel La présente norme exige l'identification d'outils et de méthodes spécifiques à utiliser pour le développement de manière générale et non pour chaque mesure de MAÎTRISE DU RISQUE.
14.7 Spécification des exigences Pour le SEMP et chacun de ses sous-systèmes (par exemple pour un sous-système SSEP), il doit exister une spécification des exigences.	**5.2 Analyses des exigences** du logiciel La présente norme traite uniquement des sous-SYSTÈMES LOGICIELS d'un SEMP.
La spécification des exigences pour un SYSTÈME ou un sous-système doit inclure et faire la distinction entre une éventuelle performance essentielle et les mesures de MAÎTRISE DU RISQUE mises en œuvre par ledit système ou sous-système.	5.2.1 Définition et documentation des exigences du logiciel d'après les exigences du SYSTÈME 5.2.2 Teneur des exigences du logiciel 5.2.3 Intégration des mesures de MAÎTRISE DU RISQUE dans les exigences du logiciel La présente norme n'exige pas que les exigences liées à des performances essentielles et à des mesures de MAÎTRISE DU RISQUE soient distinguées des autres exigences mais elle exige en effet que toutes les exigences soient identifiées de manière univoque.

Table C.3 *(continued)*

PEMS requirements from IEC 60601-1:2005	Requirements of IEC 62304 relating to the software subsystem of a PEMS
Depending on the type of product, the problem resolution SYSTEM may: – be documented as a part of the PEMS DEVELOPMENT LIFE-CYCLE; – allow the reporting of potential or existing problems affecting BASIC SAFETY or ESSENTIAL PERFORMANCE; – include an assessment of each problem for associated RISKS; – identify the criteria that must be met for the issue to be closed; – identify the action to be taken to resolve each problem.	5.1.1 Software development plan 9.1 Prepare PROBLEM REPORTS
14.6 RISK MANAGEMENT PROCESS	**7 Software RISK MANAGEMENT PROCESS**
14.6.1 Identification of known and foreseeable HAZARDS 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 NETWORK/DATA COUPLING, components of third-party origin and legacy subsystems.	7.1 Analysis of software contributing to hazardous situations This standard does not mention network/data coupling specifically
14.6.2 RISK CONTROL 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 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 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** 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. 5.2.2 Software requirements content 5.2.3 Include RISK CONTROL measures in software requirements 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.

Tableau C.3 *(suite)*

Exigences pour un SEMP de la CEI 60601-1:2005	Exigences de la CEI 62304 relatives au sous-système logiciel d'un SEMP
14.8 Architecture Pour le SEMP et chacun de ses sous-systèmes, une ARCHITECTURE doit être spécifiée et satisfaire à la spécification des exigences.	**5.3** Conception ARCHITECTURALE du logiciel
Le cas échéant, pour réduire le RISQUE à un niveau acceptable, la spécification de l'ARCHITECTURE doit utiliser: a) des composants ayant des caractéristiques de haute intégrité; b) des fonctions à sécurité positive; c) la redondance; d) la diversité; e) le découpage par fonctionnalité; f) une conception défensive, par exemple les limites des effets dangereux potentiels en réduisant la puissance de sortie disponible ou en introduisant des moyens pour limiter le déplacement des organes de manœuvre.	5.3.5 Identification des séparations nécessaires à la MAÎTRISE DU RISQUE Le découpage est la seule technique identifiée et cela uniquement parce qu'il est exigé d'indiquer la manière dont l'intégrité du découpage est assurée.
La spécification de l'ARCHITECTURE doit tenir compte: g) de l'allocation des mesures de MAÎTRISE DES RISQUEs aux sous-systèmes et éléments du SEMP; h) des types de pannes des éléments et leurs conséquences; i) des pannes ayant les mêmes causes; j) des pannes systématiques; k) de la durée des intervalles entre les essais et la couverture du diagnostic; l) de la maintenabilité; m) de la protection contre des usages abusifs raisonnablement prévisibles; n) de la spécification des liaisons réseau/données le cas échéant.	Ceci n'est pas inclus dans la présente norme.
14.9 Conception et mise en œuvre Le cas échéant, la conception doit être scindée en sous-systèmes, chaque sous-système ayant une spécification pour la conception et pour les essais.	**5.4 Conception détaillée du logiciel** 5.4.2 Elaboration de la conception détaillée de chaque UNITÉ LOGICIELLE La présente norme n'exige pas une spécification d'essai pour la conception détaillée.
Les données descriptives concernant l'environnement de la conception doivent être comprises dans le DOSSIER DE GESTION DES RISQUES.	5.4.2 Elaboration de la conception détaillée de chaque UNITÉ LOGICIELLE
14.10 VÉRIFICATION La VÉRIFICATION est exigée pour toutes les fonctions mettant en œuvre la SÉCURITÉ DE BASE, les PERFORMANCES ESSENTIELLES ou des mesures de MAÎTRISE DU RISQUE.	**5.1.6** Planification de la VÉRIFICATION du logiciel La VÉRIFICATION est requise pour chaque ACTIVITÉ.

Table C.3 *(continued)*

PEMS requirements from IEC 60601-1:2005	Requirements of IEC 62304 relating to the software subsystem of a PEMS
14.8 Architecture 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: a) COMPONENTS WITH HIGH-INTEGRITY CHARACTERISTICS; b) fail-safe functions; c) redundancy; d) diversity; e) partitioning of functionality; 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 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: g) allocation of RISK CONTROL measures to subsystems and components of the PEMS; h) failure modes of components and their effects; i) common cause failures; j) systemic failures; k) test interval duration and diagnostic coverage; l) maintainability; m) protection from reasonably foreseeable misuse; n) the NETWORK/DATA COUPLING specification, if applicable.	This is not included in this standard.
14.9 Design and implementation Where appropriate, the design shall be decomposed into subsystems, each having both a design and test specification.	**5.4 Software detailed design** 5.4.2 Develop detailed design for each SOFTWARE UNIT This standard does not require a test specification for detailed design.
Descriptive data regarding the design environment shall be included in the RISK MANAGEMENT FILE.	5.4.2 Develop detailed design for each SOFTWARE UNIT
14.10 VERIFICATION VERIFICATION is required for all functions that implement BASIC SAFETY, ESSENTIAL PERFORMANCE or RISK CONTROL measures.	5.1.6 Software VERIFICATION planning VERIFICATION is required for each ACTIVITY

Tableau C.3 *(suite)*

Exigences pour un SEMP de la CEI 60601-1:2005	Exigences de la CEI 62304 relatives au sous-système logiciel d'un SEMP
Un plan de VÉRIFICATION doit être établi pour indiquer comment ces fonctions doivent être vérifiées. Le plan doit comprendre: – Les jalons d'exécution des vérifications pour chaque fonction; – Le choix et la documentation des stratégies, ACTIVITÉs, techniques de vérification ainsi que le niveau approprié d'indépendance du personnel chargé des vérifications; – Le choix et l'utilisation des outils de vérification; – Les critères de couverture de la VÉRIFICATION.	5.1.6 Planification de la VÉRIFICATION du logiciel L'indépendance du personnel n'est pas incluse dans la présente norme. Elle est considérée prise en charge dans l'ISO 13485.
La VÉRIFICATION doit être conduite conformément au plan de VÉRIFICATION. Les résultats des ACTIVITÉs de VÉRIFICATION doivent être documentés.	Les exigences de VÉRIFICATION concernent la plupart des ACTIVITÉs.
14.11 Validation du SEMP Un plan de validation du SEMP doit inclure la validation de la SÉCURITÉ fondamentale et des performances essentielles et doit exiger des contrôles de tout fonctionnement imprévu du SEMP.	La présente norme ne couvre pas la validation du logiciel. La VALIDATION du SEMP est une ACTIVITÉ au niveau SYSTÈME et ne s'inscrit pas dans le domaine d'application de la présente norme.
La VALIDATION du SEMP doit être effectuée conformément au plan de VALIDATION du SEMP. Les résultats des ACTIVITÉs de VALIDATION du SEMP doivent être documentés.	La présente norme ne couvre pas la validation du logiciel. La VALIDATION du SEMP est une ACTIVITÉ au niveau SYSTÈME et ne s'inscrit pas dans le domaine d'application de la présente norme.
La personne entièrement responsable de la validation du SEMP doit être indépendante de l'équipe de conception. Le FABRICANT doit justifier par écrit le niveau d'indépendance.	La présente norme ne couvre pas la validation du logiciel. La VALIDATION du SEMP est une ACTIVITÉ au niveau SYSTÈME et ne s'inscrit pas dans le domaine d'application de la présente norme.
La validation d'un SEMP ne doit pas être confiée à un membre d'une équipe de conception ayant réalisé la conception du système SEMP.	La présente norme ne couvre pas la validation du logiciel. La VALIDATION du SEMP est une ACTIVITÉ au niveau SYSTÈME et ne s'inscrit pas dans le domaine d'application de la présente norme.
Toutes les relations professionnelles entre les membres de l'équipe de validation de SEMP et les membres de l'équipe de conception doivent figurer dans le DOSSIER DE GESTION DES RISQUES.	La présente norme ne couvre pas la validation du logiciel. La VALIDATION du SEMP est une ACTIVITÉ au niveau SYSTÈME et ne s'inscrit pas dans le domaine d'application de la présente norme.
Le DOSSIER DE GESTION DES RISQUES doit comporter une référence aux méthodes et résultats de la validation du SEMP.	La présente norme ne couvre pas la validation du logiciel. La VALIDATION du SEMP est une ACTIVITÉ au niveau SYSTÈME et ne s'inscrit pas dans le domaine d'application de la présente norme.
14.12 Modification Si tout ou partie d'une conception résulte de la modification d'une conception antérieure, l'ensemble des dispositions du présent paragraphe s'applique comme s'il s'agissait d'une nouvelle conception ou, la validité conservée de toute la documentation de conception préalable doit être évaluée selon une procédure documentée de modification.	**6 Processus de maintenance du logiciel** La présente norme considère par principe qu'il convient de planifier la maintenance du logiciel et qu'il est recommandé que la mise en œuvre des modifications utilise le PROCESSUS de développement du logiciel ou un PROCESSUS établi de maintenance du logiciel.

Table C.3 *(continued)*

PEMS requirements from IEC 60601-1:2005	Requirements of IEC 62304 relating to the software subsystem of a PEMS
A VERIFICATION plan shall be produced to show how these functions shall be verified. The plan shall include: – at which milestone(s) VERIFICATION is to be performed on each function; – the selection and documentation of VERIFICATION strategies, activities, techniques, and the appropriate level of independence of the personnel performing the VERIFICATION; – the selection and utilization of VERIFICATION tools; – coverage criteria for VERIFICATION.	5.1.6 Software VERIFICATION planning 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 A PEMS VALIDATION plan shall include the validation of BASIC SAFETY and ESSENTIAL PERFORMANCE, and shall require checks for unintended functioning of the PEMS.	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.
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 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** 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.

Tableau C.3 *(suite)*

Exigences pour un SEMP de la CEI 60601-1:2005	Exigences de la CEI 62304 relatives au sous-système logiciel d'un SEMP
14.13 Connexion entre le SEMP et d'autres équipements par liaisons réseau/données S'il est prévu de connecter le SEMP à un autre équipement par une liaison réseau/données qui n'est pas maîtrisée par le FABRICANT du SEMP, LA description technique DOIT: a) spécifier les caractéristiques de la liaison réseau/données nécessaire pour que le SEMP puisse satisfaire à son usage/objet prévu; b) énumérer les PHÉNOMÈNES DANGEREUX potentiels résultant d'une défaillance de la liaison réseau/données à fournir les caractéristiques spécifiées; c) informer l'organisation responsable que: – la connexion du SEMP à une liaison réseau/données qui comprend d'autres équipements pourrait entraîner des RISQUES précédemment non identifiés pour les patients, opérateurs ou tierces parties; – il convient que l'organisation responsable identifie, analyse, évalue et maîtrise ces RISQUES; – les modifications ultérieures à la liaison réseau/données pourrait introduire de nouveaux RISQUES et nécessiter une analyse supplémentaire; et – les modifications de la liaison réseau/données comprennent: ▪ les modifications de la configuration de la liaison réseau/données ▪ la connexion d'éléments supplémentaires à la liaison réseau/données ▪ la déconnexion d'éléments de la liaison réseau/données ▪ la mise à jour d'équipements connectés à la liaison réseau/données ▪ la mise à niveau d'équipements connectés à la liaison réseau/données	Les exigences relatives à la liaison réseau/données ne sont pas incluses dans la présente norme.

C.4.7 Relation avec les exigences de la CEI 60601-1-4

La CEI 60601-1-4 continuera à être utilisée jusqu'à ce que la période transitoire de la CEI 60601-1:2005 soit achevée.

Le Tableau C.4 illustre les exigences de la CEI 60601-1-4 [2] et les exigences correspondantes de la présente norme. Ceci ne signifie pas que les exigences correspondantes de la présente norme couvrent pleinement les exigences de la CEI 60601-1-4. De nombreuses parties des exigences de la norme 60601-1-4 sont couvertes par conformité à l'ISO 14971. Certaines exigences de la CEI 60601-1-4 ne sont pas traitées par la CEI 62304.

Tableau C.4 – Relation avec la CEI 60601-1-4

Exigences POUR LE SYSTÈME SEMP de la CEI 60601-1-4:1996 plus l'amendement 1:1999	Exigences correspondantes de la CEI 62304
6.8 Documents d'accompagnement	
6.8.201	4.2 et 4.3 c)
52.201 Documentation	
52.201.1	4.1
52.201.2	4.1 et 4.2

Table C.3 *(continued)*

PEMS requirements from IEC 60601-1:2005	Requirements of IEC 62304 relating to the software subsystem of a PEMS
14.13 Connection of PEMS by NETWORK/DATA COUPLING to other equipment If the PEMS is intended to be connected by NETWORK/DATA COUPLING to other equipment that is outside the control of the PEMS MANUFACTURER, the technical description shall: a) specify the characteristics of the NETWORK/DATA COUPLING necessary for the PEMS to achieve its INTENDED USE; b) list the HAZARDOUS SITUATIONS resulting from a failure of the NETWORK/DATA COUPLING to provide the specified characteristics; c) Instruct the RESPONSIBLE ORGANIZATION that: – connection of the PEMS to a NETWORK/DATA COUPLING that includes other equipment could result in previously unidentified RISKS to PATIENTS, OPERATORS or third parties; – the RESPONSIBLE ORGANIZATION should identify, analyze, evaluate and control these RISKS; – subsequent changes to the NETWORK/DATA COUPLING could introduce new RISKS and require additional analysis; and – changes to the NETWORK/DATA COUPLING include: ▪ changes in NETWORK/DATA COUPLING configuration; ▪ connection of additional items to the NETWORK/DATA COUPLING; ▪ disconnecting items from the NETWORK/DATA COUPLING; ▪ update of equipment connected to the NETWORK/DATA COUPLING; ▪ upgrade of equipment connected to the NETWORK/DATA COUPLING.	Requirements for network/data coupling are not included in this standard.

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

IEC 60601-1-4 will continue to be used until the transition period for IEC 60601-1:2005 is complete.

Table C.4 shows the requirements of IEC 60601-1-4 [2] and the related requirements in this standard. This does not indicate that the related requirements in this standard fully cover the requirements in IEC 60601-1-4. Many parts of the 60601-1-4 requirements are covered by compliance with ISO 14971. Some requirements in IEC 60601-1-4 are not addressed by IEC 62304.

Table C.4 – Relationship to IEC 60601-1-4

PEMS requirements from IEC 60601-1-4:1996 plus Amendment 1:1999	**Related requirements of IEC 62304**
6.8 Accompanying documents	
6.8.201	4.2 and 4.3 c)
52.201 Documentation	
52.201.1	4.1
52.201.2	4.1 and 4.2

Tableau C.4 *(suite)*

Exigences pour le système SEMP de la CEI 60601-1-4:1996 plus l'amendement 1:1999	Exigences correspondantes de la CEI 62304
52.201.3	4.2
52.202 PLAN DE GESTION DES RISQUES	
52.202.1	4.2
52.202.2	5.1.1,5.1.5
52.202.3	4.1, 5.1.2
52.203 Cycle de développement	
52.203.1	5.1.1
52.203.2	5.1.1
52.203.3	
52.203.4	5.1.7
52.203.5	7
52.204 Traitement de la gestion des risques	
52.204.1	4.2
52.204.2	4.2, 7
52.204.3	
52.204.3.1	
52.204.3.1.1	4.2,7.1
52.204.3.1.2	4.2,7.1.2
52.204.3.1.3	4.2
52.204.3.1.4	4.2, 7.1.2 e)
52.204.3.1.5	4.2, 7.1.2
52.204.3.1.6	4.2,7.1
52.204.3.1.7	4.2
52.204.3.1.8	4.2
52.204.3.1.9	4.2
52.204.3.1.10	4.2
52.204.3.2	
52.204.3.2.1	4.2
52.204.3.2.2	4.2,4.3
52.204.3.2.3	
52.204.3.2.4	
52.204.3.2.5	4.2
52.204.4	
52.204.4.1	4.2
52.204.4.2	4.2
52.204.4.3	4.2
52.204.4.4	4.2
52.204.4.5	
52.204.4.6	4.2

Table C.4 *(continued)*

PEMS requirements from IEC 60601-1-4:1996 plus Amendment 1:1999	Related requirements of IEC 62304
52.201.3	4.2
52.202 RISK MANAGEMENT PLAN	
52.202.1	4.2
52.202.2	5.1.1, 5.1.5
52.202.3	4.1, 5.1.2
52.203 Development life-cycle	
52.203.1	5.1.1
52.203.2	5.1.1
52.203.3	
52.203.4	5.1.7
52.203.5	7
52.204 Risk management process	
52.204.1	4.2
52.204.2	4.2, 7
52.204.3	
52.204.3.1	
52.204.3.1.1	4.2, 7.1
52.204.3.1.2	4.2, 7.1.2
52.204.3.1.3	4.2
52.204.3.1.4	4.2, 7.1.2 e)
52.204.3.1.5	4.2, 7.1.2
52.204.3.1.6	4.2, 7.1
52.204.3.1.7	4.2
52.204.3.1.8	4.2
52.204.3.1.9	4.2
52.204.3.1.10	4.2
52.204.3.2	
52.204.3.2.1	4.2
52.204.3.2.2	4.2, 4.3
52.204.3.2.3	
52.204.3.2.4	
52.204.3.2.5	4.2
52.204.4	
52.204.4.1	4.2
52.204.4.2	4.2
52.204.4.3	4.2
52.204.4.4	4.2
52.204.4.5	
52.204.4.6	4.2

Tableau C.4 *(suite)*

Exigences pour le système SEMP de la CEI 60601-1-4:1996 plus l'amendement 1:1999	Exigences correspondantes de la CEI 62304
52.205 Qualification du personnel	4.1
52.206 Spécification des exigences	
52.206.1	5.2
52.206.2	7.2.2
52.206.3	
52.207 Architecture	
52.207.1	5.3.1
52.207.2	5.3
52.207.3	
52.207.4	
52.207.5	
52.208 Conception et réalisation	
52.208.1	5
52.208.2	
52.209 Vérification	
52.209.1	5.7.1
52.209.2	5.1.5, 5.1.6
52.209.3	5.2.6, 5.3. 6, 5.4.4, 5.5.5, 5.6, 5.7
52.209.4	
52.210 Validation	
52.210.1	4.1
52.210.2	4.1
52.210.3	4.1
52.210.4	
52.210.5	
52.210.6	
52.210.7	
52.211 Modification	
52.211.1	6
52.211.2	4.1,6
52.212 Évaluation	
52.212.1	4.1

C.5 Relation avec la CEI 61010-1

Le domaine d'application de la CEI 61010-1 [4] couvre les appareils électriques d'essai, de mesurage, de régulation et de laboratoire. Seule une partie des appareils de laboratoire est utilisée dans un environnement vertical ou comme matériel de diagnostic *in vitro* (IVD).

Table C.4 *(continued)*

PEMS requirements from IEC 60601-1-4:1996 plus Amendment 1:1999	Related requirements of IEC 62304
52.205 Qualification of personnel	4.1
52.206 Requirement specification	
52.206.1	5.2
52.206.2	7.2.2
52.206.3	
52.207 Architecture	
52.207.1	5.3.1
52.207.2	5.3
52.207.3	
52.207.4	
52.207.5	
52.208 Design and implementation	
52.208.1	5
52.208.2	
52.209 Verification	
52.209.1	5.7.1
52.209.2	5.1.5, 5.1.6
52.209.3	5.2.6, 5.3.6, 5.4.4, 5.5.5, 5.6, 5.7
52.209.4	
52.210 Validation	
52.210.1	4.1
52.210.2	4.1
52.210.3	4.1
52.210.4	
52.210.5	
52.210.6	
52.210.7	
52.211 Modification	
52.211.1	6
52.211.2	4.1, 6
52.212 Assessment	
52.212.1	4.1

C.5 Relationship to IEC 61010-1

The scope of IEC 61010-1 [4] covers electrical test and measuring equipment, electrical control equipment and electrical laboratory equipment. Only part of the laboratory equipment is used in a medical environment or as in vitro diagnostic equipment (IVD).

Du fait des réglementations légales ou des références normatives, le matériel IVD est assimilé aux DISPOSITIFS MÉDICAUX sans cependant s'inscrire dans le domaine d'application de la CEI 60601-1 [1]. Ceci est imputable non seulement au fait que strictement parlant, les instruments IVD ne sont pas des DISPOSITIFS MÉDICAUX au contact direct avec les patients mais également au fait que ces produits sont fabriqués pour de nombreuses applications différentes dans divers laboratoires. L'utilisation en tant qu'appareil IVD ou en tant qu'accessoire pour appareil IVD est donc rare.

Si le matériel de laboratoire est utilisé en tant qu'appareil IVD, les résultats mesurés obtenus doivent être évalués conformément à des critères médicaux. L'application de l'ISO 14971 est exigée pour la GESTION DES RISQUES. Si ces produits comportent également des logiciels qui peuvent donner lieu à des PHÉNOMÈNES DANGEREUX, lorsque par exemple, une défaillance due au logiciel peut entraîner une modification indésirable des données médicales (résultats des mesures), la CEI 62304 doit être prise en compte.

L'organigramme de la Figure C.3 constitue une aide utile pour expliquer le principe du PROCESSUS de GESTION DES RISQUES et l'application de la CEI 62304.

Due to legal regulations or normative references, IVD equipment is allocated to MEDICAL DEVICES without, however, falling within the scope of IEC 60601-1 [1]. This is attributable not only to the fact that, strictly speaking, IVD instruments are not MEDICAL DEVICES which come into direct contact with patients, but also to the fact that such products are manufactured for many different applications in various laboratories. Use as an IVD instrument or as an accessory for an IVD instrument is then rare.

If laboratory equipment is used as IVD equipment, the measured results obtained must be EVALUATED in accordance with medical criteria. The application of ISO 14971 is required for RISK MANAGEMENT. If such products also contain software that can lead to a HAZARD, for example failure caused by the software which results in an unwanted change of medical data (measuring results), IEC 62304 must be taken into account.

The flowchart in Figure C.3 provides a useful aid to explain the principle way of the RISK MANAGEMENT PROCESS and the application of IEC 62304:

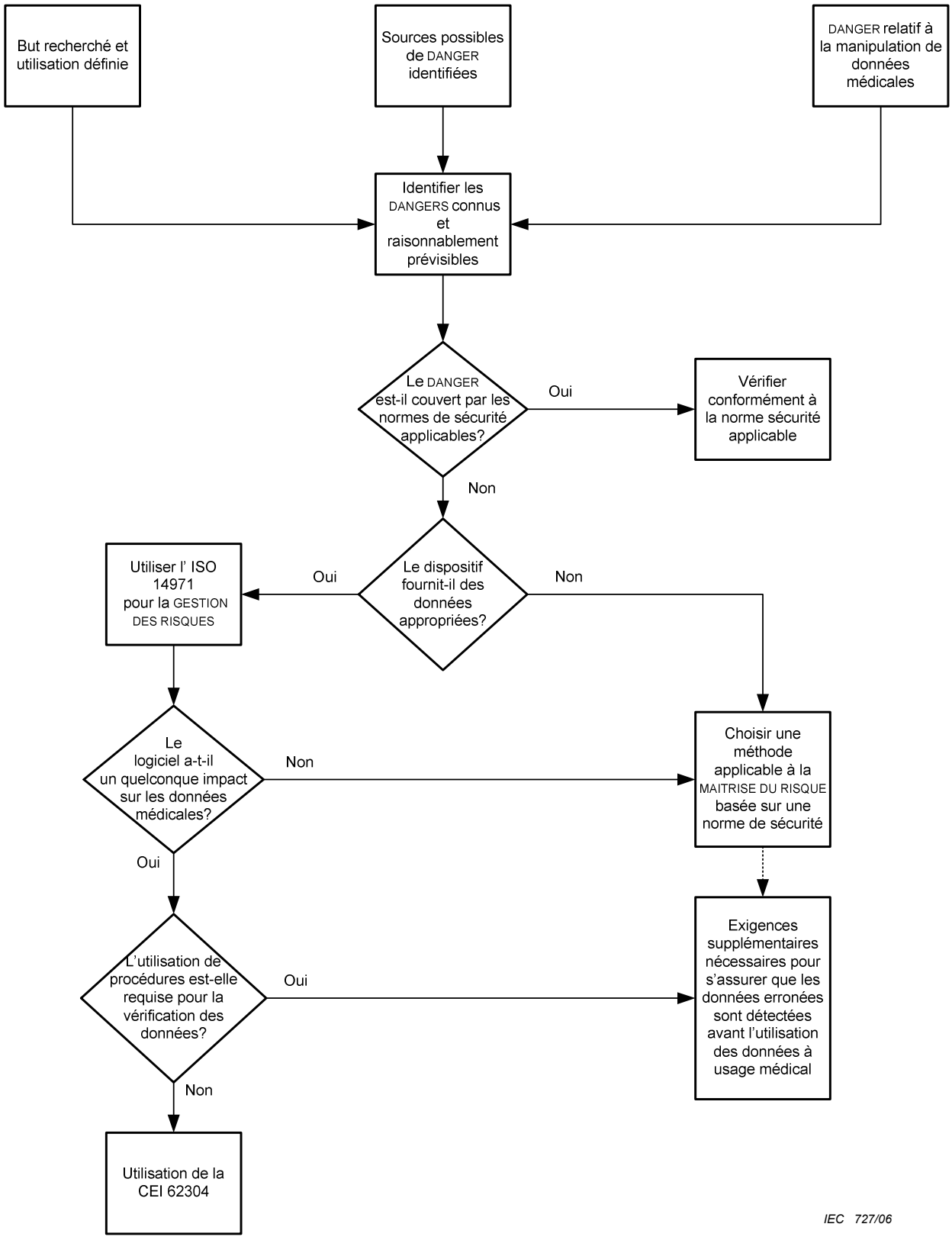

Figure C.3 – Application de la CEI 62304 avec la CEI 61010-1

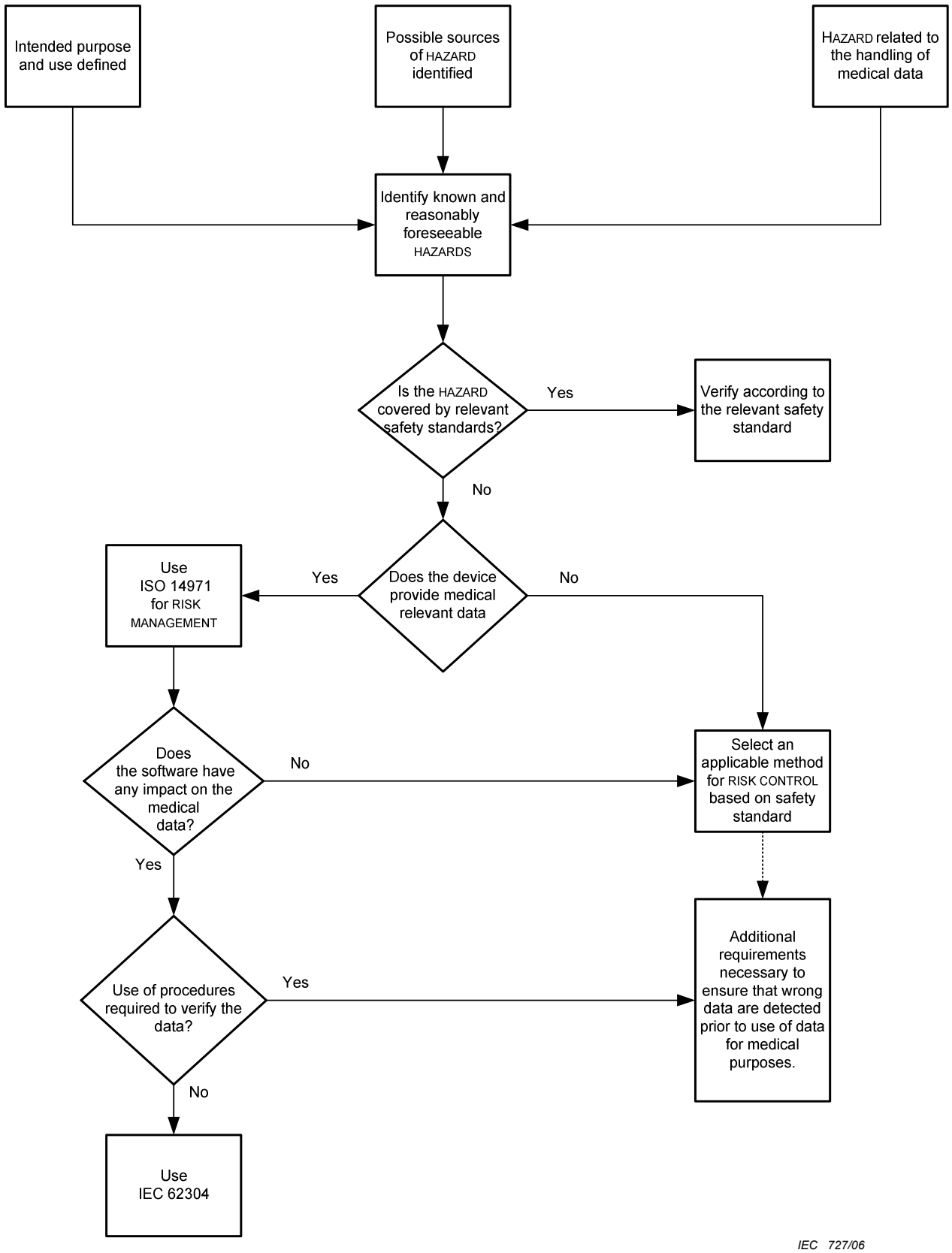

Figure C.3 – Application of IEC 62304 with IEC 61010-1

C.6 Relation avec l'ISO/CEI 12207

La présente norme est issue de l'approche et des concepts de l'ISO/CEI 12207 [9], qui définit les exigences applicables au PROCESSUS de cycle de vie des logiciels en général, c'est-à-dire sans restriction aux DISPOSITIFS MÉDICAUX.

Les principales différences de la présente norme par rapport à l'ISO/CEI 12207 sont les suivantes. Elle:

- exclue des aspects SYSTÈME tels que les exigences système, l'ARCHITECTURE et la validation système;
- omet certains PROCESSUS tels que la duplication des ACTIVITÉS qui est documentée ailleurs pour les DISPOSITIFS MÉDICAUX;
- ajoute le PROCESSUS de GESTION DES RISQUES (SÉCURITÉ) et le PROCESSUS de diffusion des logiciels;
- incorpore la documentation et la vérification qui viennent à l'appui des PROCESSUS de développement et de maintenance;
- fusionne les ACTIVITÉs de mise en œuvre et de planification de chaque PROCESSUS en une seule ACTIVITÉ dans les PROCESSUS de développement et de maintenance;
- classe les exigences par rapport aux besoins en matière de SÉCURITÉ; et
- ne classe pas explicitement les PROCESSUS comme principaux ou secondaires, ni ne les groupe comme le fait l'ISO/CEI 12207.

La plupart de ces modifications résultent du souhait d'adapter la norme aux besoins du secteur des dispositifs médicaux:

- en se concentrant sur les aspects SÉCURITÉ et la GESTION DES RISQUES du DISPOSITIF MÉDICAL de l'ISO 14971;
- en sélectionnant les PROCESSUS qui conviennent, lorsqu'ils sont utiles dans un environnement réglementé;
- en tenant compte du fait que le développement du logiciel s'intègre dans un système qualité (qui couvre certains des PROCESSUS et exigences de l'ISO/CEI 12207); et
- en réduisant le niveau d'abstraction pour en faciliter l'utilisation.

La présente norme ne comporte pas de contradiction avec l'ISO/CEI 12207. Cette dernière peut être utile comme aide à l'établissement d'un modèle de cycle de vie de développement du logiciel bien structuré qui incorpore les exigences de la présente norme.

Le Tableau C.5, qui a été préparé par le JTC1/SC7 de l'ISO/CEI, illustre la relation entre la CEI 62304 et l'ISO/CEI 12207.

C.6 Relationship to ISO/IEC 12207

This standard has been derived from the approach and concepts of ISO/IEC 12207 [9], which defines requirements for software life cycle PROCESSES in general, i.e. not restricted to MEDICAL DEVICES.

This standard differs from ISO/IEC 12207 mainly with respect to the following. It:

- excludes SYSTEM aspects, such as SYSTEM requirements, SYSTEM ARCHITECTURE and validation;
- omits some PROCESSES seen as duplicating ACTIVITIES documented elsewhere for MEDICAL DEVICES;
- adds the (SAFETY) RISK MANAGEMENT PROCESS and the software release PROCESS;
- incorporates the documentation and the VERIFICATION supporting PROCESSES into the development and maintenance PROCESSES;
- merges the PROCESS implementation and planning ACTIVITIES of each PROCESS into a single ACTIVITY in the development and maintenance PROCESSES;
- classifies the requirements with respect to SAFETY needs; and
- does not explicitly classify PROCESSES as primary or supporting, nor group PROCESSES as ISO/IEC 12207 does.

Most of these changes were driven by the desire to tailor the standard to the need of the MEDICAL DEVICE sector by:

- focusing on SAFETY aspects and the MEDICAL DEVICE RISK MANAGEMENT standard ISO 14971;
- selecting the appropriate PROCESSES useful in a regulated environment;
- taking into account that software development is embedded in a quality system (which covers some of the PROCESSES and requirements of ISO/IEC 12207); and
- lowering the level of abstraction to make it easier to use.

This standard is not contradictory to ISO/IEC 12207. ISO/IEC 12207 can be useful as an aide in setting up a well structured SOFTWARE DEVELOPMENT LIFE CYCLE MODEL that includes the requirements of this standard.

Table C.5, which was prepared by ISO/IEC JTC1/SC7, shows the relationship between IEC 62304 and ISO/IEC 12207.

Tableau C.5 – Relation avec l'ISO/CEI 12207

Processus de l'ISO/CEI 62304		Processus de l'ISO/CEI 12207	
Activité	**Tâche**	**Activité**	**Tâche**
5 PROCESSUS de développement du logiciel		5.3 Processus de développement 6.1 Processus de documentation 6.2 Processus de la gestion de la configuration 6.4 Processus de la vérification 6.5 Processus de la validation 6.8 Processus de résolution de problème 7.1 Processus de la gestion	
5.1 Planification du développement du logiciel		5.3.1 Mise en œuvre du processus 5.3.3 Conception architecturale du système 5.3.7 Codage et test du logiciel 5.3.8 Intégration du logiciel 5.3.9 Test de qualification du logiciel 5.3.10 Intégration du système 6.1.1 Mise en œuvre du processus 6.2.1 Mise en œuvre du processus 6.2.2 Identification de la configuration 6.4.1 Mise en œuvre du processus 6.5.1 Mise en œuvre du processus 6.8.1 Mise en œuvre du processus 7.1.2 Planification 7.1.3 Exécution et commande 7.2.2 Etablissement de l'infrastructure 7.2.3 Maintenance de l'infrastructure	
	5.1.1 Plan de développement du logiciel	5.3.1 Mise en œuvre du processus 7.1.2 Planification	5.3.1.1 5.3.1.3 5.3.1.4 7.1.2.1
	5.1.2 Mise à jour du plan de développement logiciel	7.1.3 Exécution et commande.	7.1.3.3
	5.1.3 Référence du plan de développement du logiciel à la conception et au développement du SYSTÈME	5.3.3 Conception architecturale du système 5.3.10 Intégration du système. 6.5.1 Mise en œuvre du processus	5.3.3.1 5.3.10.1 6.5.1.4
	5.1.4 Planification des normes, méthodes et outils de développement du logiciel	5.3.1 Mise en œuvre du processus	5.3.1.3 5.3.1.4
	5.1.5 Planification de l'intégration du logiciel et des essais d'intégration	5.3.8 Intégration du logiciel.	5.3.8.1
	5.1.6 Planification de la VÉRIFICATION du logiciel	6.4.1 Mise en œuvre du processus 5.3.7 Codage et test du logiciel 5.3.8 Intégration du logiciel 5.3.9 Test de qualification du logiciel	6.4.1.4 6.4.1.5 5.3.7.5 5.3.8.5 5.3.9.3
	5.1.7 Planification de la GESTION DES RISQUES du logiciel	Amd.1:2002 – F 3.1.5 Processus de la gestion des risques	
	5.1.8 Planification de la documentation	6.1.1 Mise en œuvre du processus	6.1.1.1
	5.1.9 Planification de la gestion de configuration du logiciel	6.2.1 Mise en œuvre du processus 6.8.1 Mise en œuvre du processus	6.2.1.1 6.8.1.1
	5.1.10 Eléments annexes à contrôler	7.2.2 Etablissement de l'infrastructure 7.2.3 Maintenance de l'infrastructure	7.2.2.1 7.2.3.1
	5.1.11 Eléments de contrôle de la configuration du logiciel avant VÉRIFICATION	6.2.2 Identification de la configuration	6.2.2.1

Table C.5 – Relationship to ISO/IEC 12207

ISO/IEC 62304 processes		ISO/IEC 12207 processes	
Activity	**Task**	**Activity**	**Task**
5 Software development PROCESS		5.3 Development process 6.1 Documentation process 6.2 Configuration management process 6.4 Verification process 6.5 Validation process 6.8 Problem resolution process 7.1 Management process	
5.1 Software development planning		5.3.1 Process implementation 5.3.3 System architectural design 5.3.7 Software coding and testing 5.3.8 Software integration 5.3.9 Software qualification testing 5.3.10 System integration 6.1.1 Process implementation 6.2.1 Process implementation 6.2.2 Configuration identification 6.4.1 Process implementation 6.5.1 Process implementation 6.8.1 Process implementation 7.1.2 Planning 7.1.3 Execution and control 7.2.2 Establishment of the infrastructure 7.2.3 Maintenance of the infrastructure	
	5.1.1 Software development plan	5.3.1 Process implementation 7.1.2 Planning	5.3.1.1 5.3.1.3 5.3.1.4 7.1.2.1
	5.1.2 Keep software development plan updated	7.1.3 Execution and control	7.1.3.3
	5.1.3 Software development plan reference to SYSTEM design and development	5.3.3 System architectural design 5.3.10 System integration 6.5.1 Process implementation	5.3.3.1 5.3.10.1 6.5.1.4
	5.1.4 Software development standards, methods and tools planning	5.3.1 Process implementation	5.3.1.3 5.3.1.4
	5.1.5 Software integration and integration testing planning	5.3.8 Software integration.	5.3.8.1
	5.1.6 Software VERIFICATION planning	6.4.1 Process implementation 5.3.7 Software coding and testing 5.3.8 Software integration 5.3.9 Software qualification testing	6.4.1.4 6.4.1.5 5.3.7.5 5.3.8.5 5.3.9.3
	5.1.7 Software RISK MANAGEMENT planning	Amd.1:2002 – F 3.1.5 Risk management process	
	5.1.8 Documentation planning	6.1.1 Process implementation	6.1.1.1
	5.1.9 Software configuration management planning	6.2.1 Process implementation 6.8.1 Process implementation	6.2.1.1 6.8.1.1
	5.1.10 Supporting items to be controlled	7.2.2 Establishment of the infrastructure 7.2.3 Maintenance of the infrastructure	7.2.2.1 7.2.3.1
	5.1.11 Software CONFIGURATION ITEM control before VERIFICATION	6.2.2 Configuration identification	6.2.2.1

Tableau C.5 *(suite)*

Processus de l'ISO/CEI 62304		Processus de l'ISO/CEI 12207	
Activité	**Tâche**	**Activité**	**Tâche**
5.2 Analyses des exigences du logiciel		5.3.3 Conception architecturale du système 5.3.4 Analyse des exigences du logiciel 6.4.2 Vérification	
	5.2.1 Définition et documentation des exigences du logiciel d'après les exigences du SYSTÈME	5.3.3 Conception architecturale du système	5.3.3.1
	5.2.2 Teneur des exigences du logiciel	5.3.4 Analyse des exigences du logiciel	5.3.4.1
	5.2.3 Intégration des mesures de MAÎTRISE DU RISQUE dans les exigences du logiciel		
	5.2.4 R Réévaluation de l'ANALYSE DU RISQUE du DISPOSITIF MÉDICAL e		Aucune
	5.2.5 Mise à jour des exigences du SYSTÈME	5.3.4 Analyse des exigences du logiciel	a) b)
	5.2.6 Vérification des exigences du logiciel	5.3.4 Analyse des exigences du logiciel 6.4.2 Vérification	5.3.4.2 6.4.2.3
5.3 Conception ARCHITECTURALE du logiciel		5.3.5 Conception architecturale du logiciel	
	5.3.1 Conversion des exigences du logiciel en ARCHITECTURE	5.3.5 Conception architecturale du logiciel	5.3.5.1
	5.3.2 Elaboration d'une ARCHITECTURE pour les interfaces d'ÉLÉMENTS LOGICIELS		5.3.5.2
	5.3.3 Spécification des exigences fonctionnelles et de performance des éléments logiciels SOUP		Aucune
	5.3.4 Spécification des matériels et des logiciels SYSTÈME nécessaires à l'élément logiciel SOUP		Aucune
	5.3.5 Identification des séparations nécessaires à la MAÎTRISE DU RISQUE		Aucune
	5.3.6 Vérification de L'ARCHITECTURE du logiciel	5.3.5 Conception architecturale du logiciel	5.3.5.6
5.4 Conception détaillée du logiciel		5.3.6 Conception détaillée du logiciel 6.4.2 Vérification	
	5.4.1 Décomposition de l'ARCHITECTURE des LOGICIELS en UNITÉS LOGICIELLES	5.3.6 Conception détaillée du logiciel	5.3.6.1
	5.4.2 Elaboration de la conception détaillée de chaque UNITÉ LOGICIELLE		
	5.4.3 Elaboration de la conception détaillée pour les interfaces		5.3.6.2
	5.4.4 Vérification de la conception détaillée	6.4.2 Vérification	5.3.6.7
5.5 Mise en œuvre et vérification des UNITÉS LOGICIELLES		5.3.6 Conception détaillée du logiciel 5.3.7 Codage et test du logiciel 6.4.2 Vérification	
	5.5.1 Mise en œuvre de chaque UNITÉ LOGICIELLE	5.3.7 Codage et test du logiciel	5.3.7.1
	5.5.2 Etablissement du PROCESSUS DE VÉRIFICATION DES UNITÉS LOGICIELLES	5.3.6 Conception détaillée du logiciel 5.3.7 Codage et test du logiciel	5.3.6.5 5.3.7.5
	5.5.3 Critères d'acceptation de l'UNITÉ LOGICIELLE	5.3.7 Codage et test du logiciel	5.3.7.5
	5.5.4 Critères supplémentaires d'acceptation de l'UNITÉ LOGICIELLE	5.3.7 Codage et test du logiciel 6.4.2 Vérification	5.3.7.5 6.4.2.5
	5.5.5 VÉRIFICATION de l'UNITÉ LOGICIELLE	5.3.7 Codage et test du logiciel	5.3.7.2

Table C.5 *(continued)*

ISO/IEC 62304 processes		ISO/IEC 12207 processes	
Activity	**Task**	**Activity**	**Task**
5.2 Software requirements analysis		5.3.3 System architectural design 5.3.4 Software requirements analysis 6.4.2 Verification	
	5.2.1 Define and document software requirements from SYSTEM requirements	5.3.3 System architectural design	5.3.3.1
	5.2.2 Software requirements content	5.3.4 Software requirements analysis	5.3.4.1
	5.2.3 Include RISK CONTROL measures in software requirements		
	5.2.4 Re-EVALUATE MEDICAL DEVICE RISK ANALYSIS		None
	5.2.5 Update SYSTEM requirements	5.3.4 Software requirements analysis	a) b)
	5.2.6 Verify software requirements	5.3.4 Software requirements analysis 6.4.2 Verification	5.3.4.2 6.4.2.3
5.3 Software ARCHITECTURAL design		5.3.5 Software architectural design	
	5.3.1 Transform software requirements into an ARCHITECTURE	5.3.5 Software architectural design	5.3.5.1
	5.3.2 Develop an ARCHITECTURE for the interfaces of SOFTWARE ITEMS		5.3.5.2
	5.3.3 Specify functional and performance requirements of SOUP item		none
	5.3.4 Specify SYSTEM hardware and software required by SOUP item		none
	5.3.5 Identify segregation necessary for RISK CONTROL		none
	5.3.6 Verify software ARCHITECTURE	5.3.5 Software architectural design	5.3.5.6
5.4 Software detailed design		5.3.6 Software detailed design 6.4.2 Verification	
	5.4.1 Refine SOFTWARE ARCHITECTURE into SOFTWARE UNITS	5.3.6 Software detailed design	5.3.6.1
	5.4.2 Develop detailed design for each SOFTWARE UNIT		
	5.4.3 Develop detailed design for interfaces		5.3.6.2
	5.4.4 Verify detailed design	6.4.2 Verification	5.3.6.7
5.5 SOFTWARE UNIT implementation and verification		5.3.6 Software detailed design 5.3.7 Software coding and testing 6.4.2 Verification	
	5.5.1 Implement each SOFTWARE UNIT	5.3.7 Software coding and testing	5.3.7.1
	5.5.2 Establish SOFTWARE UNIT VERIFICATION PROCESS	5.3.6 Software detailed design 5.3.7 Software coding and testing	5.3.6.5 5.3.7.5
	5.5.3 SOFTWARE UNIT acceptance criteria	5.3.7 Software coding and testing	5.3.7.5
	5.5.4 Additional SOFTWARE UNIT acceptance criteria	5.3.7 Software coding and testing 6.4.2 Verification	5.3.7.5 6.4.2.5
	5.5.5 SOFTWARE UNIT VERIFICATION	5.3.7 Software coding and testing	5.3.7.2

Tableau C.5 *(suite)*

Processus de l'ISO/CEI 62304		Processus de l'ISO/CEI 12207	
Activité	**Tâche**	**Activité**	**Tâche**
5.6 Intégration et essai d'intégration du logiciel		5.3.8 Intégration du logiciel 5.3.9 Essai de qualification du logiciel 5.3.10 Intégration du système 6.4.1 Mise en œuvre du processus 6.4.2 Vérification	
	5.6.1 Intégration des UNITÉS LOGICIELLES	5.3.8 Intégration du logiciel	5.3.8.2
	5.6.2 Vérification de l'intégration du logiciel	5.3.8 Intégration du logiciel 5.3.10 Intégration du système	5.3.8.2 5.3.10.1
	5.6.3 Essai du logiciel intégré	5.3.9 Essai de qualification du logiciel.	5.3.9.1
	5.6.4 Teneur des essais d'intégration		5.3.9.3
	5.6.5 Vérification des procédures d'essais d'intégration	6.4.2 Vérification	6.4.2.2
	5.6.6 Réalisation d'essais de régression	5.3.8 Intégration du logiciel	5.3.8.2
	5.6.7 Teneur de l'enregistrement des essais d'intégration	5.3.8 Intégration du logiciel	5.3.8.2
	5.6.8 Utilisation du PROCESSUS de résolution des problèmes de logiciel	6.4.1 Mise en œuvre du processus	6.4.1.6
5.7 Essais du SYSTÈME LOGICIEL		5.3.8 Intégration du logiciel 5.3.9 Essai de qualification du logiciel 6.4.1 Mise en œuvre du processus 6.4.2 Vérification 6.8.1 Mise en œuvre du processus	
	5.7.1 Etablissement d'essais pour les exigences du logiciel	5.3.8 Intégration du logiciel 5.3.9 Essai de qualification du logiciel	5.3.8.4 5.3.9.1
	5.7.2 Utilisation du PROCESSUS de résolution des problèmes de logiciel	6.4.1 Mise en œuvre du processus	6.4.1.6
	5.7.3 Contre-essais après modifications	6.8.1 Mise en œuvre du processus	6.8.1.1
	5.7.4 Vérification des essais du SYSTÈME LOGICIEL	6.4.2 Vérification 5.3.9 Essai de qualification du logiciel	6.4.2.2 5.3.9.3
	5.7.5 Teneur des enregistrements d'essai du SYSTÈME LOGICIEL	5.3.9 Essai de qualification du logiciel	5.3.9.1
5.8 Diffusion du logiciel		5.3.9 Essai de qualification du logiciel 5.4.2 Essai opérationnel 6.2.5 Évaluation de la configuration 6.2.6 Gestion de la diffusion et livraison	
	5.8.1 Assurance de l'achèvement de la VÉRIFICATION du logiciel	5.4.2 Essai opérationnel 6.2.6 Gestion de la diffusion et livraison	5.4.2.1 5.4.2.2 6.2.6.1
	5.8.2 Consignation des ANOMALIES résiduelles connues	6.2.5 Évaluation de la configuration 5.3.9 Essai de qualification du logiciel	6.2.5.1 5.3.9.3
	5.8.3 Évaluation des ANOMALIES résiduelles connues		
	5.8.4 Consignation des VERSIONS diffusées	6.2.6 Gestion de la diffusion et livraison	6.2.6.1
	5.8.5 Consignation de la manière dont le logiciel diffusé a été créé		
	5.8.6 Assurance de l'achèvement complet des ACTIVITÉS et des TÂCHES		
	5.8.7 Archivage du logiciel		
	5.8.8 Assurance de la reproductibilité du logiciel diffusé		

Table C.5 *(continued)*

ISO/IEC 62304 processes		ISO/IEC 12207 processes	
Activity	**Task**	**Activity**	**Task**
5.6 Software integration and integration testing		5.3.8 Software integration 5.3.9 Software qualification testing 5.3.10 System integration 6.4.1 Process implementation 6.4.2 Verification	
	5.6.1 Integrate SOFTWARE UNITS	5.3.8 Software integration	5.3.8.2
	5.6.2 Verify software integration	5.3.8 Software integration 5.3.10 System integration	5.3.8.2 5.3.10.1
	5.6.3 Test integrated software	5.3.9 Software qualification testing.	5.3.9.1
	5.6.4 Integration testing content		5.3.9.3
	5.6.5 Verify integration tests procedures	6.4.2 Verification	6.4.2.2
	5.6.6 Conduct regression tests	5.3.8 Software integration	5.3.8.2
	5.6.7 Integration test record contents	5.3.8 Software integration	5.3.8.2
	5.6.8 Use software problem resolution PROCESS	6.4.1 Process implementation	6.4.1.6
5.7 SOFTWARE SYSTEM testing		5.3.8 Software integration 5.3.9 Software qualification testing 6.4.1 Process implementation 6.4.2 Verification 6.8.1 Process implementation	
	5.7.1 Establish tests for each software requirement	5.3.8 Software integration 5.3.9 Software qualification testing	5.3.8.4 5.3.9.1
	5.7.2 Use software problem resolution PROCESS	6.4.1 Process implementation	6.4.1.6
	5.7.3 Retest after changes	6.8.1 Process implementation	6.8.1.1
	5.7.4 Verify SOFTWARE SYSTEM testing	6.4.2 Verification 5.3.9 Software qualification testing	6.4.2.2 5.3.9.3
	5.7.5 Document data for each test SOFTWARE SYSTEM test record content	5.3.9 Software qualification testing	5.3.9.1
5.8 Software release		5.3.9 Software qualification testing 5.4.2 Operational testing 6.2.5 Configuration evaluation 6.2.6 Release management and delivery	
	5.8.1 Ensure software VERIFICATION is complete	5.4.2 Operational testing 6.2.6 Release management and delivery	5.4.2.1 5.4.2.2 6.2.6.1
	5.8.2 Document known residual ANOMALIES	6.2.5 Configuration evaluation 5.3.9 Software qualification testing	6.2.5.1 5.3.9.3
	5.8.3 Evaluate known residual ANOMALIES		
	5.8.4 Document released VERSIONS	6.2.6 Release management and delivery	6.2.6.1
	5.8.5 Document how released software was created		
	5.8.6 Ensure activities and tasks are complete		
	5.8.7 Archive software		
	5.8.8 Assure repeatability of software release		

Tableau C.5 *(suite)*

Processus de l'ISO/CEI 62304		Processus de l'ISO/CEI 12207	
Activité	**Tâche**	**Activité**	**Tâche**
6 PROCESSUS de Maintenance du logiciel		5.5 Processus de maintenance 6.2 Processus de la gestion de la configuration	
6.1 Etablissement du plan de maintenance du logiciel		5.5.1 Mise en œuvre du processus	5.5.1.1
6.2 Analyse des problèmes et des modifications		5.5.1 Mise en œuvre du processus 5.5.2 Analyse des problèmes et des modifications 5.5.3 Mise en œuvre de la modification 5.5.5 Migration	
	6.2.1 Consignation et évaluation des retours d'information		
	6.2.1.1 Contrôle des retours d'information	5.5.1 Mise en œuvre du processus	5.5.1.1 5.5.1.2
	6.2.1.2 Consignation et évaluation des retours d'information		
	6.2.1.3 Évaluation des influences des RAPPORTS DE PROBLÈME sur la SÉCURITÉ	5.5.2 Analyse des problèmes et des modifications	5.5.2.1 5.5.2.2 5.5.2.3 5.5.2.4
	6.2.2 Utilisation du PROCESSUS de résolution des problèmes du logiciel	5.5.1 Mise en œuvre du processus	5.5.1.2
	6.2.3 Analyse des demandes de modification	5.5.2 Analyse des problèmes et des modifications	5.5.2.1
	6.2.4 Approbation des demandes de modification	5.5.2 Analyse des problèmes et des modifications	5.5.2.5
	6.2.5 Communication aux utilisateurs et aux organismes de réglementation	5.5.3 Mise en œuvre de la modification 5.5.5 Migration	5.5.3.1 5.5.5.3
6.3 Mise en œuvre de la modification		5.5.3 Mise en œuvre de la modification 6.2.6 Gestion de la diffusion et livraison	
	6.3.1 Utilisation d'un PROCESSUS établi pour mettre en œuvre la modification	5.5.3 Mise en œuvre de la modification	5.5.3.2
	6.3.2 Rediffusion du SYSTÈME LOGICIEL modifié	6.2.6 Gestion de la diffusion et livraison	6.2.6.1
7 PROCESSUS DE GESTION DES RISQUES du logiciel		Amd.1:2002 – F 3.15 Processus de la gestion des risques Le processus dans la 62304 traite des problèmes liés au risque/danger qui ne sont pas traités dans l'amendement 1. Des caractères communs existent (mesures contre le risque, etc), mais l'analyse s'oriente d'une manière totalement différente.	
8 PROCESSUS de gestion de configuration du Logiciel		5.5 Processus de maintenance 6.2 Processus de la gestion de la configuration	
8.1 Identification de la configuration		6.2.2 Identification de la configuration	
	8.1.1 Etablissement des moyens d'identification des ÉLÉMENTS DE CONFIGURATION	6.2.2 Identification de la configuration	6.2.2.1
	8.1.2 Identification des logiciels SOUP		Aucune
	8.1.3 Identification de la documentation de configuration du SYSTÈME	6.2.2 Identification de la configuration	6.2.2.1

Table C.5 *(continued)*

ISO/IEC 62304 processes		ISO/IEC 12207 processes	
Activity	**Task**	**Activity**	**Task**
6 Software maintenance PROCESS		5.5 Maintenance process 6.2 Configuration management process	
6.1 Establish software maintenance plan		5.5.1 Process implementation	5.5.1.1
6.2 Problem and modification analysis		5.5.1 Process Implementation 5.5.2 Problem and modification analysis 5.5.3 Modification implementation 5.5.5 Migration	
	6.2.1 Record and evaluate feedback		
	6.2.1.1 Monitor feedback	5.5.1 Process Implementation	5.5.1.1 5.5.1.2
	6.2.1.2 Document and EVALUATE feedback		
	6.2.1.3 Evaluate PROBLEM REPORT'S affects on SAFETY	5.5.2 Problem and modification analysis	5.5.2.1 5.5.2.2 5.5.2.3 5.5.2.4
	6.2.2 Use software problem resolution PROCESS	5.5.1 Process Implementation	5.5.1.2
	6.2.3 Analyse CHANGE REQUESTS	5.5.2 Problem and modification analysis	5.5.2.1
	6.2.4 CHANGE REQUEST approval	5.5.2 Problem and modification analysis	5.5.2.5
	6.2.5 Communicate to users and regulators	5.5.3 Modification implementation 5.5.5 Migration	5.5.3.1 5.5.5.3
6.3 Modification implementation		5.5.3 Modification implementation 6.2.6 Release management and delivery	
	6.3.1 Use established PROCESS to implement modification	5.5.3 Modification implementation	5.5.3.2
	6.3.2 Re-release modified SOFTWARE SYSTEM	6.2.6 Release management and delivery	6.2.6.1
7 Software RISK MANAGEMENT PROCESS		Amd.1:2002 – F 3.15 Risk management process Process in 62304 addresses risk / hazard issues that are not addressed in Amd 1. There is some commonality (risk measures, etc) but the focus of the analysis is quite different.	
8 Software configuration management PROCESS		5.5 Maintenance process 6.2 Configuration management process	
8.1 Configuration identification		6.2.2 Configuration identification	
	8.1.1 Establish means to identify CONFIGURATION ITEMS	6.2.2 Configuration identification	6.2.2.1
	8.1.2 Identify SOUP		none
	8.1.3 Identify SYSTEM configuration documentation	6.2.2 Configuration identification	6.2.2.1

Tableau C.5 *(suite)*

Processus de l'ISO/CEI 62304		Processus de l'ISO/CEI 12207	
Activité	**Tâche**	**Activité**	**Tâche**
8.2 Maîtrise des modifications		5.5.3 Mise en œuvre de la modification 6.2.3 Maîtrise de la configuration	
	8.2.1 Approbation des DEMANDES DE MODIFICATION	6.2.3 Maîtrise de la configuration	6.2.3.1
	8.2.2 Mise en œuvre des modifications	5.5.3 Mise en œuvre de la modification 6.2.3 Maîtrise de la configuration	5.5.3.2 6.2.3.1
	8.2.3 Vérification des modifications	6.2.3 Maîtrise de la configuration	6.2.3.1
	8.2.4 Prévision des moyens de TRAÇABILITÉ de la modification		
8.3 Documentation relative à l'état de la configuration		6.2.4 Documentation relative à l'état de la configuration	6.2.4.1
9 PROCESSUS de résolution de problème logiciel		5.5 Processus de maintenance 6.2 Gestion de la configuration 6.8 Processus de résolution du problème	
9.1 Elaboration des RAPPORTS DE PROBLÈME		6.8.1 Mise en œuvre du processus 6.8.2 Résolution du problème	6.8.1.1 b) 6.8.2.1
9.2 Etude du problème		6.8.2 Résolution du problème 6.8.1 Mise en œuvre du processus	6.8.2.1 6.8.1.1 b)
9.2 Information des parties concernées		6.8.1 Mise en œuvre du processus	6.8.1.1 a)
9.3 Utilisation du processus de la maîtrise des modifications		6.2.3 Maîtrise de la configuration. 5.5.3 Mise en œuvre de la modification	
9.4 Conservation des enregistrements		6.8.1 Mise en œuvre du processus	6.8.1.1 a)
9.6 Analyse de tendance pour les problèmes		6.8.1 Mise en œuvre du processus 6.8.2 Résolution du problème	6.8.1.1 b) 6.8.2.1
9.7 VÉRIFICATION de la résolution des problèmes du logiciel		6.8.1 Mise en œuvre du processus	6.8.1.1 d)
9.8 Teneur de la documentation d'essai			Toutes les tâches d'essais de la 12207 nécessitent une consignation

C.7 Relation avec la CEI 61508

La question a été soulevée quant à savoir s'il convient que la présente norme, qui concerne la conception de logiciels critiques pour la SÉCURITÉ, suive les principes de la CEI 61508. La position de la présente norme est expliquée ci-dessous.

La CEI 61508 traite de trois sujets principaux:

1) le cycle de vie de GESTION DES RISQUES et les PROCESSUS de cycle de vie;
2) la définition des niveaux d'intégrité de SÉCURITÉ;
3) la recommandation de techniques, d'outils et de méthodes pour le développement de logiciels et les niveaux d'indépendance du personnel chargé d'exécuter les différentes TÂCHES.

Table C.5 *(continued)*

ISO/IEC 62304 processes		ISO/IEC 12207 processes	
Activity	**Task**	**Activity**	**Task**
8.2 Change control		5.5.3 Modification implementation 6.2.3 Configuration control	
	8.2.1 Approve CHANGE REQUESTS	6.2.3 Configuration control	6.2.3.1
	8.2.2 Implement changes	5.5.3 Modification implementation 6.2.3 Configuration control	5.5.3.2 6.2.3.1
	8.2.3 Verify changes	6.2.3 Configuration control	6.2.3.1
	8.2.4 Provide means for TRACEABILITY of change		
8.3 Configuration status accounting		6.2.4 Configuration status accounting	6.2.4.1
9 Software problem resolution PROCESS		5.5 Maintenance process 6.2 Configuration management 6.8 Problem resolution process	
9.1 Prepare PROBLEM REPORTS		6.8.1 Process implementation 6.8.2 Problem resolution	6.8.1.1 b) 6.8.2.1
9.2 Investigate the problem		6.8.2 Problem resolution 6.8.1 Process implementation	6.8.2.1 6.8.1.1 b)
9.3 Advise relevant parties		6.8.1 Process implementation	6.8.1.1 a)
9.4 Use change control process		6.2.3 Configuration control. 5.5.3 Modification implementation	
9.5 Maintain records		6.8.1 Process implementation	6.8.1.1 a)
9.6 Analyse problems for trends		6.8.1 Process implementation 6.8.2 Problem resolution	6.8.1.1 b) 6.8.2.1
9.7 Verify software problem resolution		6.8.1 Process implementation	6.8.1.1 d)
9.8 Test documentation contents			All testing tasks in 12207 require documentation

C.7 Relationship to IEC 61508

The question has been raised whether this standard, being concerned with the design of SAFETY-critical software, should follow the principles of IEC 61508. The following explains the stance of this standard.

IEC 61508 addresses 3 main issues:

1) RISK MANAGEMENT life cycle and life cycle PROCESSES;
2) definition of Safety Integrity Levels;
3) recommendation of techniques, tools and methods for software development and levels of independence of personnel responsible for performing different TASKS.

Le point 1) est couvert dans la présente norme par une référence normative à l'ISO 14971 (norme du secteur des DISPOSITIFS MÉDICAUX pour la GESTION DES RISQUES). Cette référence a pour effet d'adopter l'approche de l'ISO 14971 en termes de GESTION DES RISQUES comme partie intégrante du PROCESSUS logiciel.

En ce qui concerne le point 2), la présente norme a une approche plus simple que celle de la CEI 61508. Cette dernière classe les logiciels en 4 «Niveaux d'intégrité de SÉCURITÉ» définis en termes d'objectifs de fiabilité. Les objectifs de fiabilité sont identifiés après analyse des RISQUES, quantifiant ainsi à la fois la gravité et la probabilité d'un DOMMAGE dû à une défaillance du logiciel.

La présente norme simplifie le point 2) en refusant de tenir compte de la probabilité de défaillance du logiciel avant sa classification. La classification en 3 classes de SÉCURITÉ du logiciel est fondée uniquement sur la gravité dudit DOMMAGE causé par une défaillance. Après classification, différents PROCESSUS sont exigés pour les différentes classes de SÉCURITÉ de logiciel: l'intention est de réduire encore la probabilité de défaillance du logiciel.

Le point 3) n'est pas traité par la présente norme. Ses utilisateurs sont encouragés à utiliser la CEI 61508 comme source de bonnes méthodes, techniques et outils logiciels tout en reconnaissant que d'autres approches, tant existantes que futures, peuvent fournir des résultats tout aussi bons. La présente norme ne donne pas de recommandations quant à l'indépendance des personnes chargées d'une ACTIVITÉ logicielle donnée (par exemple la VÉRIFICATION) par rapport à celles qui sont chargées d'une autre ACTIVITÉ (par exemple la conception). En particulier, il n'existe pas dans la présente norme d'exigence relative à un évaluateur de SÉCURITÉ indépendant car il s'agit d'un sujet couvert par le domaine d'application de l'ISO 14971.

Issue 1) is covered in this standard by a normative reference to ISO 14971 (the MEDICAL DEVICE sector standard for RISK MANAGEMENT). The effect of this reference is to adopt ISO 14971's approach to RISK MANAGEMENT as an integral part of the software PROCESS for MEDICAL DEVICE SOFTWARE.

For issue 2), this standard takes a simpler approach than IEC 61508. The latter classifies software into 4 "Safety Integrity Levels" defined in terms of reliability objectives. The reliability objectives are identified after RISK ANALYSIS, which quantifies both the severity and the probability of HARM caused by a failure of the software.

This standard simplifies issue 2) by disallowing consideration of probability of software failure prior to classification. Classification into 3 software safety classes is based only on the severity of that HARM caused by a failure. After classification, different PROCESSES are required for different software safety classes: the intention is to further reduce the probability of failure of the software.

Issue 3) is not addressed by this standard. Readers of the standard are encouraged to use IEC 61508 as a source for good software methods, techniques and tools, while recognising that other approaches, both present and future, can provide equally good results. This standard makes no recommendation concerning independence of people responsible for one software ACTIVITY (for example VERIFICATION) from those responsible for another (for example design). In particular, this standard makes no requirement for an independent safety assessor, since this is a matter for ISO 14971.

Annexe D
(informative)

Mise en œuvre

D.1 Introduction

La présente annexe présente la manière dont la présente norme peut être mise en œuvre dans les PROCESSUS des FABRICANTS. Elle tient également compte du fait que d'autres normes, telles que l'ISO 13485 [7] exigent des PROCESSUS appropriés et comparables.

D.2 Système de management de la qualité

Pour les FABRICANTs de DISPOSITIFS MÉDICAUX, y compris les LOGICIELS DE DISPOSITIFS MÉDICAUX dans le contexte de la présente norme, l'établissement d'un système de management de la qualité (SMQ) est exigé en 4.1. La présente norme n'exige pas que le SMQ soit nécessairement certifié.

D.3 ÉVALUATION des PROCESSUS de management de la qualité

Il est recommandé d'évaluer la manière dont les PROCESSUS des SMQ établis et documentés couvrent les PROCESSUS de cycle de vie du logiciel, en réalisant des audits, des inspections ou des analyses sous la responsabilité du FABRICANT. Il peut être remédié à toute carence identifiée en étendant les PROCESSUS des SMQ ou en les décrivant de manière séparée. Si le FABRICANT possède déjà des descriptions de PROCESSUS disponibles qui réglementent le développement, la VÉRIFICATION et la validation du logiciel, il convient également de les évaluer afin de déterminer leur conformité à la présente norme.

D.4 Intégration des exigences de la présente norme dans les PROCESSUS de management de la qualité des FABRICANTS

La présente norme peut être mise en œuvre en adaptant ou en étendant les PROCESSUS déjà mis en place dans le SMQ ou en intégrant de nouveaux PROCESSUS. La présente norme ne spécifie pas la manière dont cela doit être effectué; le choix de toute méthode convenable est laissé à la discrétion du FABRICANT.

La responsabilité du FABRICANT est de s'assurer que les PROCESSUS décrits dans la présente norme sont correctement mis en application lorsque le logiciel de DISPOSITIF MÉDICAL est développé par des équipementiers (OEM) ou des sous-traitants qui ne disposent pas de leur propre SMQ documenté.

D.5 Liste de contrôle pour les petits FABRICANTS ne disposant pas de SMQ certifié

Il convient que le FABRICANT définisse le niveau le plus élevé de classification de SÉCURITÉ du logiciel (A, B ou C). Le Tableau D.1 énumère toutes les ACTIVITÉS décrites dans la présente norme. La référence à l'ISO 13485 a pour intention d'aider à définir la place dans le SMQ. Sur la base de la classe de SÉCURITÉ du logiciel exigée, il convient que le FABRICANT évalue chaque ACTIVITÉ requise au vu des PROCESSUS existants. Si l'exigence est déjà couverte, il est recommandé de renvoyer au descriptif des PROCESSUS pertinents.

Annex D
(informative)

Implementation

D.1 Introduction

This annex gives an overview of how this standard can be implemented into MANUFACTURERS' PROCESSES. It also considers that other standards like ISO 13485 [7] require adequate and comparable PROCESSES.

D.2 Quality management system

For MANUFACTURERS of MEDICAL DEVICES, including MEDICAL DEVICE SOFTWARE in the context of this standard, the establishment of a quality management system (QMS) is required in 4.1. This standard does not require that the QMS necessarily has to be certified.

D.3 EVALUATE quality management PROCESSES

It is recommended to EVALUATE how well the established and documented PROCESSES of the QMS already cover the PROCESSES of the software life cycle, by means of audits, inspections, or analyses under the responsibility of the MANUFACTURER. Any identified gaps can be accommodated by extending the QM PROCESSES, or can be separately described. If the MANUFACTURER already has PROCESS descriptions available which regulate the development, VERIFICATION and validation of software, then these should also be EVALUATED to determine how well they agree with this standard.

D.4 Integrating requirements of this standard into the MANUFACTURER'S quality management PROCESSES

This standard can be implemented by adapting or extending the PROCESSES already installed in the QMS system, or integrating new PROCESSES. This standard does not specify how this is to be done; the MANUFACTURER is free to do this in any suitable way.

The MANUFACTURER is responsible for ensuring that the PROCESSES described in this standard are suitably put into action when the MEDICAL DEVICE SOFTWARE is developed by Original Equipment Manufacturers (OEM) or sub-contractors not having their own documented QMS.

D.5 Checklist for small MANUFACTURERS without a certified QMS

The MANUFACTURER should determine the highest software safety classification (A, B or C) of the software. Table D.1 lists all ACTIVITIES described in this standard. The reference to ISO 13485 should help to define the place in the QMS. Based on the required software safety class, the MANUFACTURER should assess each required ACTIVITY against the existing PROCESSES. If the requirement is already covered, a reference to the relevant PROCESS descriptions should be given.

En cas de divergence, il est nécessaire de prendre une mesure pour améliorer le PROCESSUS.

Cette liste peut également être utilisée pour une ÉVALUATION des PROCESSUS une fois la mesure appliquée.

Tableau D.1 – Liste de contrôle pour les petites entreprises sans SMQ certifié

ACTIVITÉ	Paragraphe correspondant de l'ISO 13485:2003	Couvert par une procédure existante ?	Si oui: Référence	Mesures à prendre
5.1 Planification du développement du logiciel	7.3.1 Planification de la conception et du développement	Oui/Non		
5.2 **Analyses** des exigences du logiciel	7.3.2 Eléments d'entrée de la conception et du développement	Oui/Non		
5.3 Conception ARCHITECTURALE du logiciel		Oui/Non		
5.4 Conception détaillée du logiciel		Oui/Non		
5.5 Mise en œuvre et vérification des UNITÉS LOGICIELLES		Oui/Non		
5.6 Intégration et essai d'intégration du logiciel		Oui/Non		
5.7 Essais DU SYSTÈME LOGICIEL	7.3.3 Résultats en sortie de la conception et du développement 7.3.4 Revue de la conception et du développement	Oui/Non		
5.8 Diffusion du logiciel	7.3.5 Vérification de la conception et du développement 7.3.6 Validation de la conception et du développement	Oui/Non		
6.1 Etablissement du plan de maintenance du logiciel	7.3.7 Maîtrise des modifications de la conception et du développement	Oui/Non		
6.2 Analyse des problèmes et des modifications		Oui/Non		
6.3 Mise en œuvre de la modification	7.3.5 Vérification de la conception et du développement 7.3.6 Validation de la conception et du développement	Oui/Non		
7.1 Analyse du logiciel en termes de contribution à des situations dangereuses		Oui/Non		
7.2 MESURES de MAÎTRISE DU RISQUE		Oui/Non		
7.3 VÉRIFICATION des mesures de MAÎTRISE DU RISQUE		Oui/Non		
7.4 GESTION DES RISQUES des modifications du logiciel		Oui/Non		
8.1 Identification de la configuration	7.5.3 Identification et TRAÇABILITÉ	Oui/Non		
8.2 Maîtrise des modifications	7.5.3 Identification et TRAÇABILITÉ	Oui/Non		
8.3 Documentation relative à l'état de la configuration		Oui/Non		
9 PROCESSUS de résolution de problème logiciel		Oui/Non		

If there is discrepancy, an action is needed to improve the PROCESS.

The list can also be used for an EVALUATION of the PROCESSES after the action has been performed.

Table D.1 – Checklist for small companies without a certified QMS

ACTIVITY	Related clause of ISO 13485:2003	Covered by existing procedure?	If yes: Reference	Actions to be taken
5.1 Software development planning	7.3.1 Design and development planning	Yes/No		
5.2 Software requirements analysis	7.3.2 Design and development inputs	Yes/No		
5.3 Software ARCHITECTURAL design		Yes/No		
5.4 Software detailed design		Yes/No		
5.5 SOFTWARE UNIT implementation and verification		Yes/No		
5.6 Software integration and integration testing		Yes/No		
5.7 SOFTWARE SYSTEM testing	7.3.3 Design and development outputs 7.3.4 Design and development review	Yes/No		
5.8 Software release	7.3.5 Design and development verification 7.3.6 Design and development validation	Yes/No		
6.1 Establish software maintenance plan	7.3.7 Control of design and development changes	Yes/No		
6.2 Problem and modification analysis		Yes/No		
6.3 Modification implementation	7.3.5 Design and development verification 7.3.6 Design and development validation	Yes/No		
7.1 Analysis of software contributing to hazardous situations		Yes/No		
7.2 RISK CONTROL measures		Yes/No		
7.3 VERIFICATION of RISK CONTROL measures		Yes/No		
7.4 RISK MANAGEMENT of software changes		Yes/No		
8.1 Configuration identification	7.5.3 Identification and traceability	Yes/No		
8.2 Change control	7.5.3 Identification and traceability	Yes/No		
8.3 Configuration status accounting		Yes/No		
9 Software problem resolution PROCESS		Yes/No		

Bibliographie

[1] CEI 60601-1:2005, *Appareils électromédicaux – Partie 1: Exigences générales pour la sécurité de base et les performances essentielles*

[2] CEI 60601-1-4:1996, *Appareils électromédicaux – Partie 1-4: Règles générales de sécurité – Norme collatérale: Systèmes électromédicaux programmables*
Amendement 1 (1999)

[3] CEI 61508-3, *Sécurité fonctionnelle des systèmes électriques/électroniques/ électroniques programmables relatifs à la sécurité – Partie 3: Prescriptions concernant les logiciels*

[4] CEI 61010-1:2001, *Règles de sécurité pour appareils électriques de mesurage, de régulation et de laboratoire – Partie 1: Prescriptions générales*

[5] ISO 9000:2005, *Systèmes de management de la qualité – Principes essentiels et vocabulaire*

[6] ISO 9001:2000, *Systèmes de management de la qualité – Exigences*

[7] ISO 13485:2003, *Dispositifs médicaux – Systèmes de management de la qualité – Exigences à des fins réglementaires*

[8] ISO/CEI 9126-1:2001, *Génie du logiciel – Qualité des produits – Partie 1: Modèle de qualité (disponible en anglais seulement)*

[9] ISO/CEI 12207:1995, *Technologies de l'information – Processus du cycle de vie du logiciel (disponible en anglais seulement)*
Amendement 1 (2002)
Amendement 2 (2004)

[10] ISO/CEI 14764:1999, *Technologies de l'information – Maintenance du logiciel (disponible en anglais seulement)*

[11] ISO/CEI 90003:2004, *Ingénierie du logiciel – Lignes directrices pour l'application de l'ISO 9001:2000 aux logiciels informatiques*

[12] ISO/CEI Guide 51:1999, *Aspects liés à la sécurité – Principes directeurs pour les inclure dans les normes*

[13] IEEE 610.12:1990, *Glossaire normalisé IEEE de la terminologie de la technologie de la programmation*

[14] IEEE 1044:1993, *IEEE standard classification for software anomalies*

[15] CEI 60601-1-6, *Appareils électromédicaux - Partie 1-6: Règles générales de sécurité - Norme collatérale: Aptitude à l'utilisation*

Bibliography

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

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

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

[4] IEC 61010-1:2001, *Safety requirements for electrical equipment for measurement, control, and laboratory use – Part 1: General requirements*

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

[6] ISO 9001:2000, *Quality management systems – Requirements*

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

[8] ISO/IEC 9126-1:2001, *Software engineering — Product quality — Part 1: Quality model*

[9] ISO/IEC 12207:1995, *Information technology – Software life cycle processes*
Amendment 1 (2002)
Amendment 2 (2004)

[10] ISO/IEC 14764:1999, *Information technology – Software maintenance*

[11] ISO/IEC 90003:2004, *Software engineering – Guidelines for the application of ISO 9001:2000 to computer software*

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

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

[14] IEEE 1044:1993, *IEEE standard classification for software anomalies*

[15] IEC 60601-1-6, *Medical electrical equipment - Part 1-6: General requirements for safety - Collateral standard: Usability*

Index des termes définis

Index of defined terms

Standards Survey

The IEC would like to offer you the best quality standards possible. To make sure that we continue to meet your needs, your feedback is essential. Would you please take a minute to answer the questions overleaf and fax them to us at +41 22 919 03 00 or mail them to the address below. Thank you!

Customer Service Centre (CSC)

International Electrotechnical Commission
3, rue de Varembé
1211 Genève 20
Switzerland

or

Fax to: **IEC**/CSC at +41 22 919 03 00

Thank you for your contribution to the standards-making process.

Nicht frankieren
Ne pas affranchir

Non affrancare
No stamp required

A Prioritaire

RÉPONSE PAYÉE

SUISSE

Customer Service Centre (CSC)
International Electrotechnical Commission
3, rue de Varembé
1211 GENEVA 20
Switzerland

Q1 Please report on **ONE STANDARD** and **ONE STANDARD ONLY**. Enter the exact number of the standard: *(e.g. 60601-1-1)*

.....

Q2 Please tell us in what capacity(ies) you bought the standard *(tick all that apply).* I am the/a:

purchasing agent ❑
librarian ❑
researcher ❑
design engineer ❑
safety engineer ❑
testing engineer ❑
marketing specialist ❑
other.....

Q3 I work for/in/as a: *(tick all that apply)*

manufacturing ❑
consultant ❑
government ❑
test/certification facility ❑
public utility ❑
education ❑
military ❑
other.....

Q4 This standard will be used for: *(tick all that apply)*

general reference ❑
product research ❑
product design/development ❑
specifications ❑
tenders ❑
quality assessment ❑
certification ❑
technical documentation ❑
thesis ❑
manufacturing ❑
other.....

Q5 This standard meets my needs: *(tick one)*

not at all ❑
nearly ❑
fairly well ❑
exactly ❑

Q6 If you ticked NOT AT ALL in Question 5 the reason is: *(tick all that apply)*

standard is out of date ❑
standard is incomplete ❑
standard is too academic ❑
standard is too superficial ❑
title is misleading ❑
I made the wrong choice ❑
other .....

Q7 Please assess the standard in the following categories, using the numbers:
(1) unacceptable,
(2) below average,
(3) average,
(4) above average,
(5) exceptional,
(6) not applicable

timeliness.....
quality of writing.....
technical contents.....
logic of arrangement of contents .....
tables, charts, graphs, figures.....
other .....

Q8 I read/use the: *(tick one)*

French text only ❑
English text only ❑
both English and French texts ❑

Q9 Please share any comment on any aspect of the IEC that you would like us to know:

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Enquête sur les normes

La CEI ambitionne de vous offrir les meilleures normes possibles. Pour nous assurer que nous continuons à répondre à votre attente, nous avons besoin de quelques renseignements de votre part. Nous vous demandons simplement de consacrer un instant pour répondre au questionnaire ci-après et de nous le retourner par fax au +41 22 919 03 00 ou par courrier à l'adresse ci-dessous. Merci !

Centre du Service Clientèle (CSC)

Commission Electrotechnique Internationale
3, rue de Varembé
1211 Genève 20
Suisse

ou

Télécopie: **CEI**/CSC +41 22 919 03 00

Nous vous remercions de la contribution que vous voudrez bien apporter ainsi à la Normalisation Internationale.

Nicht frankieren
Ne pas affranchir

Non affrancare
No stamp required

A Prioritaire

RÉPONSE PAYÉE

SUISSE

Centre du Service Clientèle (CSC)
Commission Electrotechnique Internationale
3, rue de Varembé
1211 GENÈVE 20
Suisse

Q1 Veuillez ne mentionner qu'**UNE SEULE NORME** et indiquer son numéro exact: (*ex. 60601-1-1)*

.....

Q2 En tant qu'acheteur de cette norme, quelle est votre fonction?
(cochez tout ce qui convient)
Je suis le/un:

agent d'un service d'achat ❑
bibliothécaire ❑
chercheur ❑
ingénieur concepteur ❑
ingénieur sécurité ❑
ingénieur d'essais ❑
spécialiste en marketing ❑
autre(s).....

Q3 Je travaille:
(cochez tout ce qui convient)

dans l'industrie ❑
comme consultant ❑
pour un gouvernement ❑
pour un organisme d'essais/ certification ❑
dans un service public ❑
dans l'enseignement ❑
comme militaire ❑
autre(s).....

Q4 Cette norme sera utilisée pour/comme
(cochez tout ce qui convient)

ouvrage de référence ❑
une recherche de produit ❑
une étude/développement de produit ❑
des spécifications ❑
des soumissions ❑
une évaluation de la qualité ❑
une certification ❑
une documentation technique ❑
une thèse ❑
la fabrication ❑
autre(s).....

Q5 Cette norme répond-elle à vos besoins:
(une seule réponse)

pas du tout ❑
à peu près ❑
assez bien ❑
parfaitement ❑

Q6 Si vous avez répondu PAS DU TOUT à Q5, c'est pour la/les raison(s) suivantes:
(cochez tout ce qui convient)

la norme a besoin d'être révisée ❑
la norme est incomplète ❑
la norme est trop théorique ❑
la norme est trop superficielle ❑
le titre est équivoque ❑
je n'ai pas fait le bon choix ❑
autre(s) .....

Q7 Veuillez évaluer chacun des critères ci-dessous en utilisant les chiffres
(1) inacceptable,
(2) au-dessous de la moyenne,
(3) moyen,
(4) au-dessus de la moyenne,
(5) exceptionnel,
(6) sans objet

publication en temps opportun .....
qualité de la rédaction.....
contenu technique .....
disposition logique du contenu .....
tableaux, diagrammes, graphiques, figures.....
autre(s) .....

Q8 Je lis/utilise: *(une seule réponse)*

uniquement le texte français ❑
uniquement le texte anglais ❑
les textes anglais et français ❑

Q9 Veuillez nous faire part de vos observations éventuelles sur la CEI:

.....
.....
.....
.....
.....
.....

ISBN 2-8318-8637-6

ICS 11.040

Typeset and printed by the IEC Central Office
GENEVA, SWITZERLAND