Frequently Asked Questions related to the Implementation of EN 62304:2006 with respect to MDD 93/42/EEC

Version: 1.0

Date: April 5, 2013

<empty page>

Table of Contents

<empty page>

Introduction

Aim of the FAQ 62304

The international standard IEC 62304 ("MEDICAL DEVICE software – Software life-cycle processes") provides requirements for the development and maintenance of medical software. Published in 2006, it covers software, both embedded in MEDICAL DEVICEs and as a MEDICAL DEVICE. In Europe, the -technically identical- EN 62304 version is a harmonized standard under all three MEDICAL DEVICEs directives:
AIMDD, 90/385/EEC; MDD, 93/42/EEC; and IVDD, 98/79/EC.

This document aims to clarify questions that relate to the use of EN 62304:2006 in the context of the European MEDICAL DEVICEs Directives. It also intends to provide guidance on technical and regulatory matters relevant for application of the standard. Finally, this document also aims to be a reference for medical software manufacturers, as well as for Notified Bodies dealing with medical software. Although this document has been reviewed by a voluntary team consisting of a few NBs, the aim is that it should be used by all NBs as a reference document to ensure more consistent application of the standard.

Rationale

In recent years, many questions have arisen concerning how certain elements of the standard need to be understood in the context of the European MEDICAL DEVICE regulatory framework. Experts from European Notified Bodies and European MEDICAL DEVICE industry started to request and collect these questions.

Questions submitted, numbering well over one-hundred, have been sorted and categorized. Some questions showed overlap, others could be combined. Eventually, 73 unique questions remained divided into seven categories. Answers were prepared by the drafting team, and reviewed by the IEC/ISO group which developed IEC 62304 and some European Notified Bodies.

Drafting team

The drafting team consisted of the following people:

Jomuna Choudhuri, VDE Test and Certification Institute
Koen Cobbaert, Quality, Regulatory and Risk Management, Agfa Healthcare
Georg Heidenreich, Quality & Technology, Siemens AG - Healthcare Sector
Frans Jacobs, Regulatory Affairs manager X-ray products, Philips Healthcare
Gerd Neumann, Software Standardization Expert, Siemens AG - Healthcare Sector
Michael Bothe, Head of Medical devices/Processes/Systems, VDE Test & Certification Institute
Peter Linders, Chair Technical & Regulatory Affairs Committee, COCIR

Comments on this FAQ may be submitted to: FAQ62304@vde.com. We realize that this FAQ is neither perfect nor complete. Depending on the comments we receive on this FAQ, or on other developments related to implementation of IEC 62304 in Europe, we may decide to update or amend this publication.

<empty page>

1 Abbreviations

Words written in SMALL CAPS are defined terms. Their definition can be found in the “Terms and Definitions” section of IEC 62304

AIMDD	Active Implantable MEDICAL DEVICE Directive
CMS	Configuration Management System
COCIR	European Coordination Committee of the Radiological, Electromedical and Healthcare IT Industry
COTS	Commercial off-the-shelf
EEC	European Economic Community
EUROM VI	European Federation of Precision Mechanical and Optical Industries – Medical Technology
FPGA	Field Programmable Gate Array
GPO	General Practitioner’s Office
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
IVDD	In-Vitro Diagnostic Directive
MDD	MEDICAL DEVICE Directive
MEDDEV	Non-binding guidance for MEDICAL DEVICES, endorsed by EU Member States http://ec.europa.eu/health/medical-devices/documents
MPBetreibV	Medizinprodukte – Betreiberverordnung Verordnung über das Errichten, Betreiben und Anwenden von Medizinprodukten (Medical Devices Operator Ordinance The regulations governing the setting up, operation, use and maintenance of medical devices) Relevant only for Germany
NB, NBs	Notified Body, Notified Bodies
PEMS	Programmable Electrical Medical System
SAAS	Software as a service
SDD	Software Detailed Design,
SIL	Safety Integrity Levels as per IEC 61508
SOUP	Software of Unknown Provenance
TÜV	Technischer Überwachungsverein (Technical Inspection Association)
VDE	Verband der Elektrotechnik, Elektronik und Informationstechnologien (Association for Electrical, Electronic and Information Technologies)

<empty page>

2 Questions and Answers

2.1 Scope of EN 62304

2.1.1 Does EN 62304 relate to only the MDD (93/42/EEC)?

Answer:

No, the standard has been harmonized under all three medical devices directives but for simplicity only the MDD is mentioned in this document.

2.1.2 When is software considered a MEDICAL DEVICE?

Answer:

See MEDDEV 2.1/6 (chapter 2).

2.1.3 How does the standard distinguish between open and closed systems?

Answer:

There is no differentiation in the standard between closed or open systems.

2.1.4 Assuming all software has a medical purpose, does the standard apply to the following?

a) SAAS

b) Embedded software including FPGA's with single chip computers

c) Hardware Description Languages specifying FPGAs

d) Stand alone

e) Medical apps

f) Excel macros

g) Open and closed systems

h) Internet or cloud based

i) Server based systems

j) Network devices

Answer:

If the intended use qualifies the software as a MEDICAL DEVICE or if the software is part of a MEDICAL DEVICE, all of the above are within the scope of the standard, as long as such software can be executed during the intended operation. Notes:

a) SAAS

In some case, the service provided is not only the use of the software but can also include various additional services for instance:

- Data storage capability
- Medical expertise/decision
- …

MDD does not cover the overall service provided.

MDD only covers design, manufacturing and regulatory post market activities of the medical devices.

Nevertheless, it is the responsibility of the MD legal manufacturer of the software intended to be used as part of a wider service to manage the specific risks related to the use of the software itself under the service environment.

b) Embedded software including FPGA's with single chip computers

Software executed on a processor (can also be part of a FPGA) during the intended operation is considered a software item under EN 62304.

c) Hardware Description Languages specifying FPGAs

Specifications (e.g. in some Hardware Description Language) to be executed during production of the FPGA are considered tools and do not fall under the term medical device and are not SW items in the sense of IEC 62304.

d) Stand alone

Since 2007 the MDD considers software not intended to be used specifically for incorporation into a physical medical device as an independent medical device in its own right, provided its intended use includes medical purposes.

e) Medical apps

Despite its easy availability and easy installation, apps with an intended medical use fall under the MDD and must be created according to EN 62304.

f) Excel macros

Excel macros sold with an intended medical use fall under the MDD and must be created according to EN 62304.

However, if the clinician creates own macros or modifies existing ones this work is under the **MPBetreibV** if used in Germany. In other Member States, other requirements may apply.

g) Open and closed systems

See question 2.1.3

h) Internet or cloud based

i) Server based systems

j) Network devices

An internet based, server based or cloud based software that meets the definition of the MDD is a medical device. Any general purpose operating system or network software is a SOUP. Any general purpose commercially available hardware devices such as network or storage capability that does not meet the definition of an accessory according MDD are only non-medical components. Nevertheless, risk associated with such HW architecture has to be managed in the medical device risk management file.

2.1.5 Can a manufacturer get a process and a product certification based on EN 62304**?**

Answer:

Some notified bodies provide services relating to EN 62304 and even issue “private” certificates which do not fall under a specific accreditation yet. Therefore, such a certification is not mandatory.

2.1.6 What information is the EN 62304 providing in regard to the life-cycle management of medical devices incorporated into an IT medical network?

Answer:

It is not providing any information related to IT medical networks because EN 62304 applies to software in a MEDICAL DEVICE or to software as a MEDICAL DEVICE in its own right.

2.1.7 Does EN 62304 cover all requirements in the General Principles of Software Validation (as published by FDA) for product software?

Answer:

EN 62304 does not cover software validation. It is intentionally left outside of the scope of the standard. As for embedded software, PEMS validation is a system level activity and thus is covered in chapter 14 of EN 60601-1 (3rd. Ed.). The future IEC 82304 will cover validation of software-only products (standalone software). A less direct link to validation for these products is triggered in EN-ISO 13485:2012 because this standard (although not mandatory under EN 62304 (see clause 4.1)), also sets requirements for design and development validation in clause 7.3.6.

The FDA guidance uses the term “validation” to mean the sum total of verification activities - which are covered by EN 62304 - and the subsequent validation that the verified software satisfies its user needs and intended use.

2.1.8 As validation and final release are not included in EN 62304, which standard provides the requirements for these activities such that compliance with the MDD can be achieved / proven?

Answer:

For embedded software, validation is covered in chapter 14 of EN 60601-1 (3rd. ed.) within the context of the entire system. For standalone medical software, no current standard covers the validation aspects of the essential requirements of the Directive.

However, manufacturers of stand-alone software who apply quality management standards such as EN ISO 13485 have to fulfill the validation requirements of that standard.

2.1.9 What are the expectations of the Notified Bodies in regard to EN 62304 Compliance?

Answer:

Compliance with EN 62304 gives the *presumption of conformity* with some of the essential requirements of the Directive. If the standard is not applied, the manufacturer has to provide other objective evidence showing the software is in conformance with the corresponding essential requirements. Although the application of the standard is voluntary, it represents the current state of the art and as such shall be used by the Notified Body as a frame of reference for assessing the objective evidence supplied by the manufacturer.

2.1.10 Is tailoring of the standard allowed when only some degree of compliance can be claimed?

Answer:

The software as a product must comply with the applicable essential requirements of the directive. EN 62304 can be used to support the claim of compliance with the applicable directive.

Tailoring is not allowed from the perspective of "degree of compliance"; however, depending on the safety classification of the software, the standard adapts the requirements regarding the extent of content and documentation needed (less for class A software). Nevertheless, if compliance to EN 62304 is claimed, full compliance needs to be achieved for all applicable clauses.

2.1.11 Will my organization need a full re-assessment once a new version of the standard is published?

Answer:

It depends on the changes in the second edition of IEC 62304 and (with regard to the requirements of the MDD) on whether the second edition is harmonized, superseding the first one.

2.1.12 Class A Software.

While I recommend using EN 62304 also for a "true" Class A software, don't you think that the status of Harmonized standard with regulatory impact for Class A is a constraint because by definition "No injury or damage to health is possible"?

Answer:

No, it represents the minimal set of activities and tasks which should be performed when developing and/or maintaining medical software to demonstrate that it is really class A software. During the life cycle it may be necessary to update the risk analysis and possibly reclassify the software.

2.1.13 Should we expect an update to EN 62304 now that IEC 60601-1-4 (PEMS) was rolled into IEC 60601-1 Clause 14?

Answer:

Although the revision of IEC 62304 is in progress and publication is expected in 2015, this change within the IEC 60601 domain was not one of the causes for the revision of IEC 62304. Therefore this change in the IEC 60601 domain will not lead to changes in the revised IEC 62304.

2.1.14 What is the purpose of creating IEC 82304?

Answer:

The main aim of IEC 82304 is to cover product related requirements for software-only products, such as validation and labeling in a single product standard.

2.1.15 The naming of MEDICAL DEVICE Software, Health Software and Healthcare software are not easy to understand. Which type of software follows under each category? Please provide a table with definition of these three different categories.

Answer:

EN 62304 uses only the term Medical Device Software (clause 3.12). Definitions for the other terms are being developed (see for example IEC/CD 82304).

2.2 Placing Software as MEDICAL DEVICE on the Market

2.2.1 Is EN 62304 alone sufficient to fulfill the Essential Requirements of the MDD for a standalone software product?

Answer:

No. Compliance with EN 62304 does not provide a presumption of conformity with all applicable essential requirements of Annex I of the MEDICAL DEVICE Directive. EN 63204 for instance does not cover usability aspects, clinical evaluation, and the final validation of the software product or the need for accompanying documents such as user instructions. Therefore, other standards and procedures need to be considered to show complete fulfillment of all applicable essential requirements. (If harmonized standards are not applied, the manufacturer has to justify and explicitly state the selected equivalent alternative methods)

2.2.2 Instead of going through all this hassle with EU guidelines and conformity, I prefer writing in the intended use of my software that it should not be used for diagnosis or therapeutics. Is this OK? I mean, otherwise I cannot compete with my Apps with other developers

Answer:

Your claim is your responsibility. Be careful with your intended use statement. If your product is used by many as a medical device, and your product clearly has features that allow it to be used as such, you may be held liable for the off-label use of your software. In addition, if your product does not fall under the MDD, it is likely to fall under other regulations that may have more stringent safety requirements, e.g. GPSD (General Product Safety Directive)

See also MEDDEV 2.1/6 (chapter 4 Modules)

2.2.3 Conformity assessment procedure for software as MEDICAL DEVICE:

a) Can software as MEDICAL DEVICE (standalone software) of class IIb or III be assessed based on Annex III+V of the MDD or Annex III+IV only?

b) What is the Notified Body procedure during an audit of Annex II.3 to investigate if a manufacturer has implemented the requirements of IEC 62304?

Answer:

a) According to article 11 of the directive, it is allowed for medical devices of class IIb or class III to use either the Annex II route, Annex III plus Annex IV, or Annex V. However, the MDD may not take all peculiarities of medical software into account, and type examination is not really considered appropriate.

b) QMS audits, in particular Annex II.3 audits, are performed to determine compliance with Annex II.3 of the directive. It is not the intention of such audits to check the compliance with a standard like EN 62304.
The NB can take some samples during the audit to make a plausibility check if the application of EN 62304 is not only claimed but also applied.
But the manufacturer cannot derive full compliance with EN 62304 from audit results.

2.2.4 Is IEC 62304 accepted / required in other regions / countries [for the] regulatory approval process?

Answer:

It is very likely that there is similar acceptance of IEC 62304 in other countries. For example it is recognized by the FDA under recognition number 13-8 and has been translated into an identical Chinese standard YY/T 0664.

2.2.5 We do have our requirements in a requirement management tool, and the designs are in an architecture modeling tool. Now, the question is whether we have to generate and sign off something like ".pdf" out of the tools or if it is sufficient to keep and baseline the data in the respective repositories? What would be the conditions to maintain the electronic form only?

Answer:

EN 62304 requires formal approval of change requests (see clause 6.2.4 and 8.2.1) and on top of that the Quality Management System (see clause 4.1) according to e.g. ISO 13485 in which the software life-cycle processes are embedded will require that documents are controlled. There are many ways and probably even more tools to control documents. Signing off on ".pdf" documents can be one of them.

See also question 2.3.3 and 2.3.4

2.2.6 Classification of software as MEDICAL DEVICE:

a) Is there any relation between the safety classification according to EN 62304 and the classification of the MDD, Annex IX?

b) For software that is embedded in a medical device, how does the classification of the device influence the classification of the software according to EN 62304?

Answer:

a) No, regulation and the standard do not describe a mapping between safety class and MDD classification which has to be derived by interpreting the intended use.

b) There is no direct influence

2.2.7 How is compliance with EN 62304 confirmed by NBs?

Are those NBs accredited for certifying this compliance?

Answer:

Full compliance with EN 62304 cannot be demonstrated by a Notified Body system audit (ISO 9001/ISO 13485/Annexes of the directives) under ISO/IEC 17021 accreditation because Notified Bodies assess systems and documents to show compliance with directives.

Testing laboratories can demonstrate full compliance with EN 62304 either by assessing product specific documents (under an ISO/IEC 17025 accreditation) or by a product independent process audit, which certifies the compliance of software life-cycle-processes in general. The laboratories then issue either a private certificate (see question 2.1.5) or a certificate under the accreditation of ISO/IEC 17065.

2.2.8 Can a manufacturer comply with EN 62304 by having a quality management system in place that is not certified?

Answer:

EN 62304 does not require a specific quality management system. However, it is required that, according to clause 4.1, the "manufacturer of MEDICAL DEVICE software shall demonstrate the ability to provide MEDICAL DEVICE Software that consistently meet customer requirements and applicable regulatory requirements". This can be demonstrated by a quality management system, which does not necessarily need to be certified.

2.3 Life-cycle Processes

2.3.1 If software development is an outsourced activity, what is expected from the Notified Body as evidence that the service supplier's software development process is in compliance with EN 62304?

Answer:

The NB expects the manufacturer to be in control of the service supplier. For compliance with EN 62304, the service supplier must have the processes in place and have produced all the **documents** required by EN 62304 for those processes that have been outsourced. The manufacturer should clearly define the activities and tasks to be performed by the supplier as well as the activities performed by the manufacturer in which the supplier is involved.

For example, if the code development and unit testing have been outsourced, the service supplier should provide evidence of those activities, the manufacturer must do the remaining activities, such as integration, etc..

2.3.2 The development of the software is outsourced to a software developer who is not certified to EN ISO 13485, neither to EN 62304, nor to EN ISO 14971.

What other regulations would the software developer need to adhere to?

Answer:

EN 62304, EN ISO 14971 and EN ISO 13485 are standards, not regulations. In the end, it is the manufacturer who has to comply with the MDD requirements. It is up to the manufacturer and their suppliers how they share the burden of establishing the necessary compliance evidence, preferably expressed in a contractual agreement between manufacturer and supplier.

See also question 2.3.1.

2.3.3 Does this standard have an equivalent expectation to requirements such as those addressed in FDA Part 11 (Electronic Records & Signatures) in the US?

Answer:

Although EN 62304 does not require a specific quality management system, this standard has been tailored to be implemented under a QMS. A system according EN ISO 13485 requires that the documents are controlled. FDA's 21 CFR part 11 is explicit when it comes to how documents must be controlled. FDA's 21 CFR part 11 becomes applicable when premarket clearance for the USA is requested and **IEC 62304** related information is sent to FDA electronically.

2.3.4 What kind of review process should be applied on Requirement, Design and Test Specifications at the end of each iteration when updated versions are available?

Is there any formal sign off needed?

Answer:

The manufacturer has freedom to define the review and approval process. EN 62304, however, requires that these processes are appropriate to the scope, complexity and software safety classification of the Software System to be developed. In particular change requests require formal approval.

EN ISO 14971 requires the maintenance of documents related to Risk Management. In addition, the quality system EN ISO 13485 also requires control of documents.

See for example clauses 5.1.8, 5.2.6, 5.5.2, 6.2.4, 8.2.1, and 9.4, Annex B and table C.3 of EN 62304.

2.3.5 Does EN 62304 require a specific development process?

Answer:

No, the manufacturer has the freedom to establish a software development process. EN 62304**,** however, requires that these processes are appropriate to the scope, complexity and software safety classification of the software system to be developed.

See clause 5.1.1.

2.3.6 Why is EN 62304 not organized around deliverables?

Answer:

EN 62304 is a process standard and is organized around activities. It gives you the freedom to organize your deliverables and tailor them to the needs of your specific development processes. However, be careful: many clauses contain requirements for deliverables.

Especially clause 5.1 makes it very clear that the deliverables must be planned.

2.3.7 Couldn't a manufacturer implement the processes 5 to 9 at a "project level" and ensure that software development and maintenance considers customer and regulatory requirements?

Answer:

Yes, the processes described in clauses 5 to 9 can be implemented at a "project level" but is has to be kept in mind that the project cannot end until the end-of-life of the product.

2.3.8 Are there any restrictions for dividing up the requirements/responsibilities of EN 62304 between a manufacturer and a software subcontractor that should be adhered to?

Answer:

Not really, almost anything can be delegated to the subcontractor. However, there are restrictions such as:

Clause 6.2.1 Document and evaluate feedback

Clause 6.2.4 Change request approval

Clause 6.2.5 Communicate to users and regulators

But the manufacturer has the final responsibility over the software system.

See also question 2.3.1

2.3.9 How does EN 62304 map against TickIT Plus?

Answer:

TickIT is about the application of EN ISO 9001 to software development and not specific to MEDICAL DEVICES.

2.3.10 How do the maintenance activities in EN 62304 relate to ISO 20000/ITIL?

Answer:

ISO/IEC 20000 & ITIL deal with life cycle Service Management and are larger process frameworks compared to EN 62304, but they do not contradict each other.

Maintenance activities within EN 62304 are from the manufacturer point of view once a MEDICAL DEVICE has been released, while ISO/IEC 20000-1 & ITIL look at the maintenance in the context of overall service management. Due to this difference in focus, one has to be aware that the EN 62304 focuses on patient and user risk management, defining more "preventive" maintenance actions rather than the more "corrective" approach found within general IT.

2.3.11 What are the artifacts required by EN 62304?
We came up with the following list. A summary list and the applicable EN 62304 section would be very helpful.

- **Software Development Plan**
- **Software Architecture document**
- **Software Requirements Specification document(s)**
- **Software Detailed Design document(s)**
- **Software Unit Test Specification document(s)**
- **Software Integration Test Specification document(s)**
- **Software Regression Test Specification document(s)**
- **Software Unit Test Report document(s)**
- **Software Integration Test Report document(s)**
- **Software Regression Test Report document(s)**
- **Software Configuration Management Plan?**

Answer:

The standard requires following documents:

- Risk Management File (clause 4.2, 7)
- Software Safety Classification (clause 4.3.c)
- Software Development Plan (clause 5.1.1)
- Software System requirements (5.2), including risk control measures (clause 5.2.3)
- Software Architectural Design (clauses 5.3, 5.4)
- Software Test Plan (clauses 5.5, 5.6, 5.7, especially 5.7.1 NOTE 1 and 2)
- Traceability Overview (of test procedures to software requirements) (clause 5.7.4)
- Software Test Report (clause 5.7.5)
- Residual Anomalies (clause 5.8)
- Configuration Management (clauses 5.8.4, 5.8.5, 8)

2.3.12 At what level does the Problem Resolution Process apply?

Problem resolution can occur during the formal Design Verification phase before a software release to the field. During this phase, testing reveals anomalies that need to be tracked and evidence needs to be provided that the anomaly was fixed. Problem resolution also occurs after the software is in the field. Large problems found in the field can trigger an immediate software release with a fix and smaller problems can be scheduled to be fixed in the next software release. Generally, problem resolution at this level is specified as part of the QMS and is much broader than software. At what level does Chapter 9 apply? We assumed only at the Design Verification level but a consultant implied it also applied to the field level. We need some clarification.

Answer:

This is a life cycle standard, meaning that the problem resolution process is not only applicable to the development of a software system but also to maintenance of a released software system.

See for example clauses 5.1.1 e), 5.6.8, 5.7.2, 6.1 d) and 6.2.2 of EN 62304

See Annex 1- Figure 1 in this FAQ document

2.3.13 Does software refactoring require a formal change request?

Frequently areas of the software are refactored to repay what is known in the industry as "technical debt". This refactoring improves the codebase for the future but is not associated with a defect or a new feature. The standard doesn't really address these types of changes. Our conclusion is these changes are documented in the change control system so the appropriate unit tests and integration tests are run but these changes are not triggered by a formal change request because they originated from within the software development group. We believe this fits with EN 62304 because B.8.2 allows CHANGE REQUESTS to be made by a technical lead. The problem is making every single change to the software require a CHANGE REQUEST is totally impractical and would create impediments to improving the code base for the future.

Answer:

Definition of refactoring: Improving the software, or reducing technical debt, without changing behavior or functionality. In other words, the end result/output of the software stays the same, but how the result is produced is changed or clarified (see reference [6]).

Yes, refactoring requires a formal change request. From the moment you start testing and integrating, your configuration management has to start. This includes formal change control. It's up to the manufacturer to determine the granularity of the change request.

See also question 2.3.4

2.3.14 What information should be included in the technical file to show compliance with EN 62304?

Answer:

The technical file should provide enough information about the processes, activities and tasks applied during the development or maintenance of the software.

The technical file should also provide information about the deliverables (see list of documents from question 2.1.11) which were generated by using the processes mentioned in the technical file.

2.3.15 How can agile processes be EN 62304 compliant?

Answer:

EN 62304 does not prescribe a specific software development process.

As a result, agile processes can be done in an EN 62304-compliant way. Simply put, EN 62304 only requires activities and documents. The activities can be performed in an incremental fashion and then be iterated. The documents have to be consistent and managed under configuration management.

See also reference [6] as a valuable source of additional information.

2.4 Risk Assessment and Risk Management

2.4.1 Clause 7.2.1 requires software items with a safety class B or C to have defined and documented risk control measures for each potential contributing cause for a hazardous situation. Is it acceptable that risk control measures cover several causes at once, rather than creating a risk control measure for each individual cause?

Answer:

While EN 62304 clause 7.2.1 requires risk control measures for each potential cause, it is possible to cover multiple causes by a single risk control measure. In fact, separate risk control measures for each individual cause may lead to complex and, therefore, less safe software. In the end, it is the overall risk mitigation that counts.

2.4.2 When and why can the safety class of a SOFTWARE SYSTEM be reduced?

Answer:

The safety class of a software system can only be reduced by one level (C to B and B to A) by means of hardware risk control (see clause 4.3.a).

As to the why, this clause assumes that a hardware risk control is capable to either reduce the consequence or the probability of a failure in such a way that the risk becomes acceptable.

It has to be stressed that a safety class can only be reduced if the hardware risk control measure is successful in mitigating the risk to an acceptable level (see Clause 4.3a).

2.4.3 How shall ISO 14971 be used together with EN 62304?

Answer:

EN 62304 prescribes in clause 4.2 the use of a RISK MANAGEMENT PROCESS complying with ISO 14971 for all safety classes. For those parts implemented as software, EN 62304 defines some process requirements.

To achieve safety and effectiveness of software in/as MEDICAL DEVICE, it has to be proven that the software fulfills the specifications without causing unacceptable risks. In ISO 14971, the risk for medical devices is detailed at system level, and EN 62304 requires compliance with it. Building on ISO 14971, EN 62304 focuses on guidelines specific to software in chapter 7.

We would like to reference IEC/TR 80002-1 as a valuable source of additional information.

2.4.4 How does reduction of probability affect the required activities and the software safety class under EN 62304?

Answer:

The software safety class is independent of probabilities.

For each identified chain of events, the contribution of software to a hazard is assumed to be 100% probability. However, reducing the probability can be part of an effective and appropriate mitigation.

2.4.5 Explain Hazard, Cause, Sequence of Events in the context of software

Answer:

A hazard is abstract, and hazardous situations are instances (manifestations) of hazards. So, a hazardous situation is a hazard manifested as a real event.

Cause: Initial event, resulting in a sequence or combination of events, eventually contributing to a Hazard

Examples for sequence of events:

Example for software embedded in hardware:

1.	Condition:	Patient unattended on table, unattended object near control unit falls on the control unit and presses the table up button
2.	Hazard:	Uncontrolled motoric movements of the patient table
3.	Hazardous situation:	Patient stuck between table and X-ray device
4.	Harm:	Thorax contusion of patient between patient table and X-ray device

Example for software only product:

1.	Condition:	Dataset from a database is imported
2.	Hazard:	During processing the software flips the image
3.	Hazardous situation:	The doctor mistakes the laterality of the body part
4.	Harm:	The doctor amputates the wrong leg

2.4.6 When should we expect additional Software Hazard Analysis guidance within EN 62304?

Answer:

Conducting a safety assessment is detailed in ISO 14971; additional guidance can be found in IEC/TR 80002-1.

2.5 Classification and Segregation

2.5.1 What is segregation?

Answer:

Segregation is a way to ensure that software items do not influence each other in an unintended way.

Segregation means setting apart or separating things. Segregation is intended to avoid side-effects resulting from dependencies of control flow, data flow and shared resources.

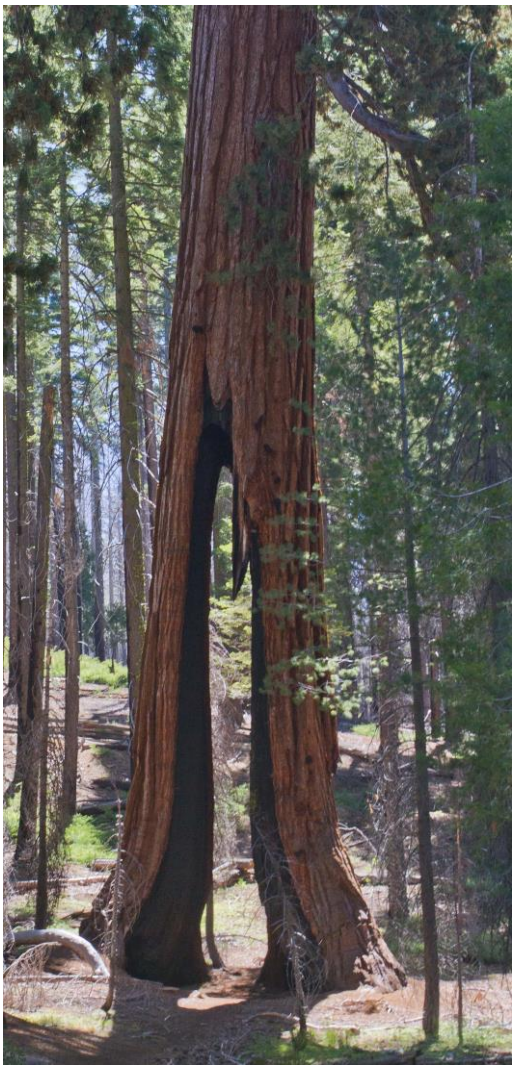
It works on three different levels: functional, logical and physical.

Functional segregation can for example be established via middle ware or 'wrappers'. They prevent the use of SOUP (see chapter 2.7 of this FAQ) features you do not want your system to use.

Physical segregation can involve separate processors.

Logical segregation can involve separate memory allocation.

The type of segregation needed depends on the elements of your system that may pose a critical failure state.



2.5.2 How do I prove segregation to be effective?

Answer:

Segregation aims to avoid unintended side-effects between software items from dependencies of control flow, data flow and shared resources.

Your proof of segregation is effective by demonstrating that there are no significant side effects.

Example of segregation:

In most cases, distinct operating system processes are appropriate to segregate class-C-items from other items - since operating systems intend to segregate processes.

For each class C-unit, the critical resources should be determined. A reliable measure is to claim required resources during startup of each class C-unit.

CPU-time - if that is a critical resource - can be ensured via process priorities or multiple processors or even multiple CPU boards.

Common approach:

1. Design and construction measures establish segregation
2. Perform safety analytic techniques, like FTA (Fault tree analysis) and FMEA (Failure Mode and Effect Analysis).
3. Verification will prove that segregation is effective.

Verification must demonstrate that the use of resources (physical or time) by the safety-related software item is appropriate to avoid unintended impact with the execution environment (other processes running on the same box). If test cases in the lab show that there is low performance and invalid measures are taken to hastily speed up the software, then these measures possibly negatively impact the design and add other risks through unforeseen side-effects.

"Specify segregation so verification can demonstrate that (under foreseeable operation conditions) the verification segregation is effective. Consider specifying the following:

- *Data flow corruption is prevented: non-safety related software items cannot modify safety-related data*
- *Control flow corruption is prevented: safety-related functions can always execute at the correct time, without being effected by the actions of the non-safety-related software items*
- *Non-safety-related software items cannot modify the safety-related software items*
- *Corruption of the execution environment is prevented: corruption of parts of the software system used by both safety-related and non-safety-related software items (e.g. processor registers, device registers and memory access privileges) cannot occur."*

See also reference [1]

Verification may also focus on the availability of shared resources, e.g. by creating a stress situation while examining the proper function of C-units.

2.5.3 If a class-B-software uses COTS (like e.g. the run-time library of a compiler), which criteria must be fulfilled for a sufficient separation of the class-B-software from class-A-COTS? Or is COTS allowed under these circumstances only if it is developed according to class-B-process or higher? Or is a validation of a class-A-COTS possible, and according to which criteria?

Answer:

By definition 3.29 all COTS (off-the-shelf software) are SOUP.

In EN 62304 the clauses 5.3.3, 5.3.4 state specifications requirements, clause 5.3.6 describes the need to verify SOUP operation. There are no explicit requirements to segregate SOUP. There are no assumptions on how SOUP has been developed. It is, however, important to verify (clause 5.3.6) SOUP according to its intended use within the software architecture - as specified in clauses 5.3.3 and 5.3.4.

In practice this means specifying how SOUP shall be working and to implement a sufficient set of representative test cases for SOUP. In the run-time library example this could mean writing extra software that uses the required features and tests the results in explicit way.

Note that SOUP may have a safety classification (A, B, C), however, EN 62304 does not raise specific requirements depending on such a safety class.

2.5.4 How does the severity under the intended use relate to the software safety class?

Answer:

EN 62304 requires you to start with assigning a C safety class. However, it is to be assumed that the intended use of the device is very clear before any safety assessment can be started, which leads to the safety classification of the device and the software items.

ISO 14971 helps to determine how software is part of a chain of events that potentially contributes to hazards.

Every chain of events which has been identified by the manufacturer as a contributor to a hazard under reasonable circumstances must be addressed.

The intended use of the device must be very clear before you start a safety assessment in order to determine the software safety classification.

During the safety assessment you identify and analyze each chain of events that can lead to a risk to health under reasonable circumstances.

If a chain of events can lead to serious injury, then the software is class C. If it cannot lead to serious injury (used in accordance with its intended use) then it is class B. If no injury can result, the safety class will be A.

Subsequent hardware control measures that significantly lower the risk can reduce the safety class by one level (from C to B OR from B to A). Safety class reduction is not possible through user information (such as training or safety notices in the manual) because the outcome is reviewed by a doctor as these are not hardware risk control measures.

When refining software items, the child items inherit the safety class of the parent item by default, unless the manufacturer has documented a rationale that the refined item cannot contribute to hazards with the same severity (see clause 4.3.d). Then the software class of the child item can be lower than the safety class of the containing software item.

The combination of severity and probability determine the acceptability of residual risks. See ISO 14971.

Without serious injury the product (under its intended use) is B or lower.

Without any injury, the safety class will be A.

Subsequent hardware control measures - significantly lowering the risk - can reduce the safety class by one level (from C to B OR from B to A).

Safety Class reduction is not possible through user information (such as training) or professional review by a doctor - as these are not hardware control measures.

When refining software items the child items inherit the safety class of the parent item. For refined items and units not contributing to hazards, the software class can be lower than the safety class of its containing software item.

ISO 14971 and probability help to determine the "acceptability" of residual risks.

2.5.5 There is no difference in the level of design control if software items cannot be architecturally segregated. For development of such monolith software the determination of the software safety class for each software item adds no value.

Can we claim compliance to EN 62304 if our procedures make clause 4.3 (assigning a software safety class) optional, i.e. dependent on the desire of the project team to use different levels of design control for the different software items?

Answer:

No. Assigning a Software Safety Class is compulsory, not optional. It is, however, permissible to limit this to an initial classification of the whole software system.

2.5.6 EN 62304 must be applied to the complete MEDICAL DEVICE (consisting of a medical control system and a protective system). The control system becomes primarily class C based on the probability of death or serious injury. By introducing an independent protective system the classification of the control system does not change because the protective system is not purely HW (it contains embedded SW).

The protective system becomes class C because of probability of death or serious injury.

We have Class C and C: Is this interpretation correct?

Answer:

This rationale given in the question is in contradiction to clause 7.2.2b of EN 62304, if the protective system is implemented as risk control related to the control system.

According to clause 7.2.2 the protective system has to be classified as C. This means that: The assigned software safety class defines the rigor of the software processes which must be applied to the risk control item. In this case, it is irrelevant if the protective system probably never causes death or serious injury.

Downgrading safety class is only allowed with subsequent and pure HW protection, so indeed the classification of the Control system remains class C. The complete MEDICAL DEVICE (control and protection) will remain class C because the protective system is not a pure hardware risk control.

2.5.7 Does a class B software generated by a compiler imply class B also to the compiler?

Which criteria exist for a sufficient separation of the class-B-software from a class-A-compiler?
Or for a sufficient validation of a class-A-compiler to generate class-B-software?
Which documentation is required from the supplier to ensure the compiler's compliance with class B?

Answer:

Tools need not be safety classified but must be validated (see ISO 13485 clause 7.5.2.1).

It has to be noted that re-distributable components of a compiler (e.g. runtime libraries) are SOUP of the MEDICAL DEVICE.

2.5.8 How are development platforms and tools related to the software safety class?

Answer:

Development platforms and tools are not considered medical software; therefore no safety class needs to be assigned.

Only medical device software (according cl. 1.2 of EN 62304) and its parts have to be safety classified. Development platforms and other tools are not classified as they do not fall under cl. 1.2 of EN 62304.

2.5.9 What is the relation between the Risk Analysis at System level and the Software Safety Classes?

Answer:

The software safety class of a Software System gives an indication of the overall contribution of the severity of risks that are associated to the use of the Software System. This overall contribution is based on a Risk Analysis at Software System level. The safety class sets the strictness of the process requirements for the development and maintenance of the software system.

2.5.10 The 3 safety classifications in EN 62304 seem to be very similar to the 3 levels of concern defined by the FDA. Please explain how they differ, if at all.

Answer:

The software *safety classification* in EN 62304 is an instrument to define the strictness of the development and maintenance processes in advance. The software *level of concern* is an instrument to define software deliverables which have to be included in a regulatory submission. One could say that the required deliverables (FDA) lead indirectly to processes which should have been followed to accommodate the submission.

There is some correlation but in general the regulatory classification is independent of the assignment of risk class in EN 62304. The software safety class depends only on risk severity and does not take into account likelihood of harm or probability. The *level of concern* is an aggregate estimate of the complete risk posed to patients exposed to the device and certainly incorporates these factors of risk.

Although the wording in the definitions is slightly different, we believe that the levels are identical with respect to severity of HARM only. So IEC's A, B and C can be correlated with FDA's Minor, Moderate, and Major levels of concern.

EN 62304 allows software classes to be changed, according to clause 4.3, while FDA graduation cannot be changed.

2.5.11 How do you correlate IEC 61508 SIL levels to EN 62304 safety classifications?

Answer:

Since Safety Classes are determined at analysis time and before assessing the impact of mitigations and because SILs are one element in reducing the assessed risk, only a Risk Analysis at system level can establish a relation between SIL and Safety Classes.

2.5.12 Can the Software Safety Class be listed in the Software Architecture instead of the Risk Management File? Does the Risk Management File have a statement pointing back to the Software Architecture?

Section 4.3c states the safety class assigned to each SOFTWARE SYSTEM goes in the risk management file (also implied by 7.2.2b). There is a possibility that changes in the Software Architecture will affect the classification and go undetected. We prefer to keep the safety classification in the Software Architecture and have the Risk Management File pointing to the Software architecture. This will minimize risk of changes in the software architecture causing an undetected change in the safety classification. This should be stated as being allowed in the FAQ.

Answer:

The standard requires safety classification but it does not specify a document in which this should be done. So, documenting the safety class in the Software Architecture document or even a separate document is allowed. It is up to the manufacturer to determine how it wants to document the safety classification. Be aware that the document in which the safety class is documented is part of the Risk Management File.

2.5.13 Software Classification is a real issue with big impacts.

Notified body auditors use the word "indirectly" in the Serious Injury definition to conclude more or less for all related diagnosis information that majority of such software are in Class C. You can always theoretically find a very improbable scenario (far of the current medical practice) but as Classification is only linked with the consequences it is argued that it is Class C

In addition, very often Class B is chosen (even the notion of "NON serious injury" is not very relevant) because:

You can demonstrate that Serious Injury is not possible according to claims and intended use.

You may have difficulties to say for a MEDICAL DEVICE that "No injury or damage to health is possible" as, for example, at least: slight delay for treatment (without urgent situation) or repetition of an exam (without X-ray dose)

Would it be possible for clarification:

a) To define the meaning of the word "indirectly". In my understanding, "indirectly" is associated with a time issue/urgency situation as an alarm of a monitor which is not functional and could lead indirectly to a serious injury or the death if no actions are taken by medical staff?

b) To give some examples of software for each class to provide clues for helping classification, in particular for Class A and B (Class C there is no problem!)?

Answer:

In the current 1st edition of IEC 62304 “indirectly” is not defined in relation to SERIOUS INJURY, nor has it been defined in ISO 14971. For the moment our advice is to interpret directly versus indirectly as per the Joint COCIR EUROM VI position paper on direct diagnosis (14 October 2011)

See Annex 4 of this FAQ-Document.

2.6 Specifications, testing and tools

2.6.1 I'm a manufacturer of MEDICAL DEVICES which consist of hardware and embedded software.

How do I document my requirements and tests?

Do I need to split my documents?

Answer:

There is no formal requirement to split documents, however, experience shows that it is very practicable to split them into hardware and software related documents.

2.6.2 My question is about Web-based medical software. Imagine a software installed on a server in the manufacturer's facility and some doctors have password and username to enter this software via web to access treatment calculations. Does 62304 have specific requirements related to digital certificates,(http or https?) server requirements, server room requirements?

Answer:

EN 62304 is a process standard that describes activities and documents for producing evidence. It does not raise specific product requirements. At the time of writing, there is no product standard for medical software. (See also chapter 2.2)

2.6.3 EN 62304 is about the life–cycle process.

How about device specific software requirements (non-process related)?

Answer:

This standard describes the software development process, including the deliverables which are device specific. So, the software requirements for a specific device are documented in the software requirements specification for the specific device (clause 5.2).

2.6.4 There is a circular dependency between risk analysis and functional specs, i.e. the risk analysis is based on the features described in the functional spec on one hand, on the other hand the risk analysis will provide input to the functional specs in form of mitigations.

So, how to resolve this situation?

Should we have a released version of the functional spec first and a second review after the mitigations are defined?

Answer:

It is not a circular dependency but rather an iterative process.

It is up to the manufacturer to define the starting point and approach.

2.6.5 Requirements and Design Input

What is the appropriate level of granularity of requirements as design input?

Is the requirement specification enough or do we need formally reviewed functional specifications?

Answer:

EN 62304 does not prescribe a certain granularity of requirements or software units. Requirements should be testable by criteria which produce "accepted" or "not accepted" results. For commercially-sized systems it is recommended to document a Functional specification and to split the SOFTWARE-SYSTEM into items and units.

2.6.6 Design description

What is the appropriate level of granularity for design descriptions?

Is the architecture spec enough or do we need formally reviewed detailed component design specifications?

Maybe for safety relevant code only?

Answer:

Clause 5.3 requires architecture for software in class B or C.

Clause 5.4.2 says that for class C the detailed design is required.

2.6.7 Which (if any) of the tracing requirements are meant to be bi-directional?

Answer:

The standard only requires traceability in the clauses 5.1.1.c, 7.3.3 and 8.2.4. Respectively at system level (Class A, B, C), risk management level (Class B, C) and change control level (Class A, B, C). There is no explicit requirement for bi-directionality. Of course, the standard does not forbid having bi-directional traceability: it may be helpful to have an overview that the tests performed fully cover of all the requirements.

For an overview of the dependencies which need to be traced, see Annex 3 of this FAQ-Document.

2.6.8 DEPLOYMENT

Installation carrier (medium): For the installation of a class-B-software via media (e.g. DVD) and networks (e.g. the Internet), are there special means required beyond the standard techniques to ensure that the image of the installed class-B-software is identical to the source image?

Are check programs for this purpose to be classified as B, and which criteria are to be fulfilled, e.g. for the reliability of a check sum?

Answer:

Tools for the development, deployment or maintenance of software don not inherit the safety class from the product they are used with. Tools are classless. As such, a runtime compiler is classless, (except in the rare case that the compiler is part of the MEDICAL DEVICE). Nevertheless, the standard requires tools to be controlled when used with software items of class B or C. It is up to the manufacturer to validate the use of a tool for its intended purpose (Ref. ISO 13485, clause 7.5.2.1). Validation of tools (and for that matter the validation criteria needed for the reliability of a check sum) are outside the scope of this standard.

2.6.9 Coherence between requirements

According to 5.4.2, for Class B Software Units:

Documented SDD are not required for all Units

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]

However, § 5.5.5 states:

5.5.5 SOFTWARE UNIT VERIFICATION

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

How can you document Software Unit Verification without formal documented SDD for all software units? Could you give some examples to clarify this point, or is it allowable that for Class B, Software unit verification can be done indirectly by Software item (which includes Software units) tests?

Simplified question:

The standard requires me to document software unit verification of class B items (§ 5.5.5), but it does not require me to document the detailed design specifications of class B items. How can I verify against something that I did not have to document?

Answer:

Absence of documentation does not mean it does not exist. The detailed design specification is in the minds of the developers and testers. For class B items that is considered sufficient for a unit test. Testing against an undocumented specification implies that you would not report on the detailed steps performed during the unit test, but that you would merely list the software item and conclude with a pass or fail. For class C items detailed design specifications are required allowing you to document your unit testing in more detail.

Also read clauses 5.5.3 and 5.5.4 carefully; unit acceptance criteria are often a subset of the unit verification so these clauses may suggest additional verification methods.

2.6.10 Software development and testing uses tools and objects found in shared open sources (forums) where verification is unlikely.

Does this standard set precedence for control of such open source tools?

What activities or documents are required by EN 62304 for such tools?

Answer:

The standard also applies to open source code. If you take the code it follows the requirements of a SOUP. If you make changes to the code, then you must consider it as a software item that you developed yourself. The level of control depends on the safety class of the code.

2.6.11 External source in unit test tool:

What activities / documents are required by EN 62304 if a unit test tool has been developed using source code from an externally available library?

Answer:

Test tools have to be evaluated and are part of the CMS.

See clauses 5.1.4, 5.5.2, and 5.8.8, as well as ISO 13485, clause 7.5.2.1.

2.7 SOUP and Legacy Software

2.7.1 How do I assess and qualify suppliers of SOUP (Software of Unknown Provenance) software? When the software has not been specifically developed for incorporating into a MEDICAL DEVICE.

Answer:

EN 62304 does not have specific requirements on SOUP suppliers other than the general supplier management requirements (EN 62304 clause 4.1 and e.g. EN ISO 13485). For SOUP items, the following specific requirements apply clauses: 5.1.7, 5.3.3, 5.3.4, 5.3.6, 6.1, 7.1.2, 7.1.3, 8.1.2.

For more information see Annex 2 of this FAQ-Document.

2.7.2 This standard acknowledges the existence of SOUP (Software of Unknown Provenance). What rigor of testing and documentation would SOUP require to meet requirements for EN 62304?

Answer:

Except for the detailed development (clause 5.4) and the software coding activity (clause 5.5) SOUPs require the same activities as software items you develop yourself.

2.7.3 What in EN 62304 prevents a manufacturer from declaring all their software is SOUP, whether it is or not, and doing less work?

Answer:

Nothing really prevents the manufacturer from declaring all the software to be SOUP provided it meets the definition of SOUP (clause 3.29). Note that it is not necessarily less work: even more work may be required before the product can be placed on the market.

Except for the detailed development (clause 5.4) and the software coding activity (clause 5.5.) SOUPs require the same tasks as software items you develop yourself.

There are also the additional tasks specified by EN 62304 which are specific to SOUPs (e.g. SOUP monitoring). If you are into keeping up appearances the supplier selection activities for the SOUP (selection, certification and determination of critical or non-critical supplier) all require additional effort, which in the end may not have been worth it.

Looking at it from the MDD perspective, the manufacturer gets into problems when he has to show compliance to the essential requirements. E. g.: Essential Requirement 2:
“The solutions adopted by the manufacturer for the design and construction of the devices must conform to safety principles, taking account of the generally acknowledged state of the art. (...)”

The solutions of the manufacturer would not be state of the art.

See Annex 2 Figure 2.

2.7.4 If a software (either stand-alone or embedded) was designed prior the publication of EN 62304 but it is still being placed on the market (legacy software):

a) Do we need to do something?

b) If yes, what?

Answer:

a) Yes.

b) Assuming it is in the context of MEDICAL DEVICE software, it would need to meet the current "State of the art". You can decide to:

- Bring it into compliance with EN 62304 immediately
  (In this case demonstration of compliance to **EN 62304** is possibly not sufficient for marketing the software. The requirements of the MDD require possibly further changes to the software and the technical file).
  or
- Follow as will be proposed in Annex E of the future EN 62304 Ed 2:

The intent of this Annex is to create a baseline of your legacy software based on the information which is available from sources such as

- Post market information from the use of this device
- The documentation you have available from your development process and the outcome of a gap analysis on what is missing in relation to EN 62304

After having collected all the information, a decision can be made how to proceed.

One of your decisions could be that you have enough information to comply with the standard. If not you can decide to create all of the appropriate information to meet the standard, starting with classifying your software and continue with other information such as:

- Risk analysis, requirements, architecture, design, implementation and tracing.
- The build process, integration and the testing activities can be repeated. The testing activities will create new test records.

Presupposition:

A configuration management must exist (versioning and reproduction of build environment and SOUP).

Be aware that if you change the legacy software, those changes need to follow the entire standard.

More advanced guidelines will become available in the proposed Annex on legacy software in the second edition of IEC 62304.

2.7.5 If legacy software needs to be significantly changed, what processes and documents are required to achieve/maintain compliance with EN 62304? And when are changes considered to be significant?

Answer:

It depends on what information is available about the legacy software. For full compliance, all processes and documents required must be considered.

EN 62304 does not consider the significance of changes. Any change requires you to consider possible effects or implications of these changes to your product. The output of this assessment will determine the relevant activities to be performed.

References

[1] *David A. Vogel (30 November 2010). Medical Device Software Verification, Validation, and Compliance. Artech House. pp. 8–. ISBN 978-1-59693-422-1.*

[2] *MEDDEV 2.1/6 (January 2012): Guidelines on the qualification and classification of stand alone software used in healthcare within the regulatory framework of medical devices*

[3] *IEC 62304 Ed. 2.0 is currently in preparation and is expected 2014 – 2015*

[4] *IEC 82304 Ed. 1.0 is currently in preparation and is expected 2015*

[5] *COCIR/EUROM VI Position Paper on Direct Diagnosis 14 October 2011*

[6] *AAMI TIR 45 Guidance of the use of AGILE practices in the development of medical device software 2012*

<empty page>

Annex 1 Software Problem Resolution Process

There are several entry points to the problem resolution process, both during development and maintenance of the software (relates to question 2.3.12).

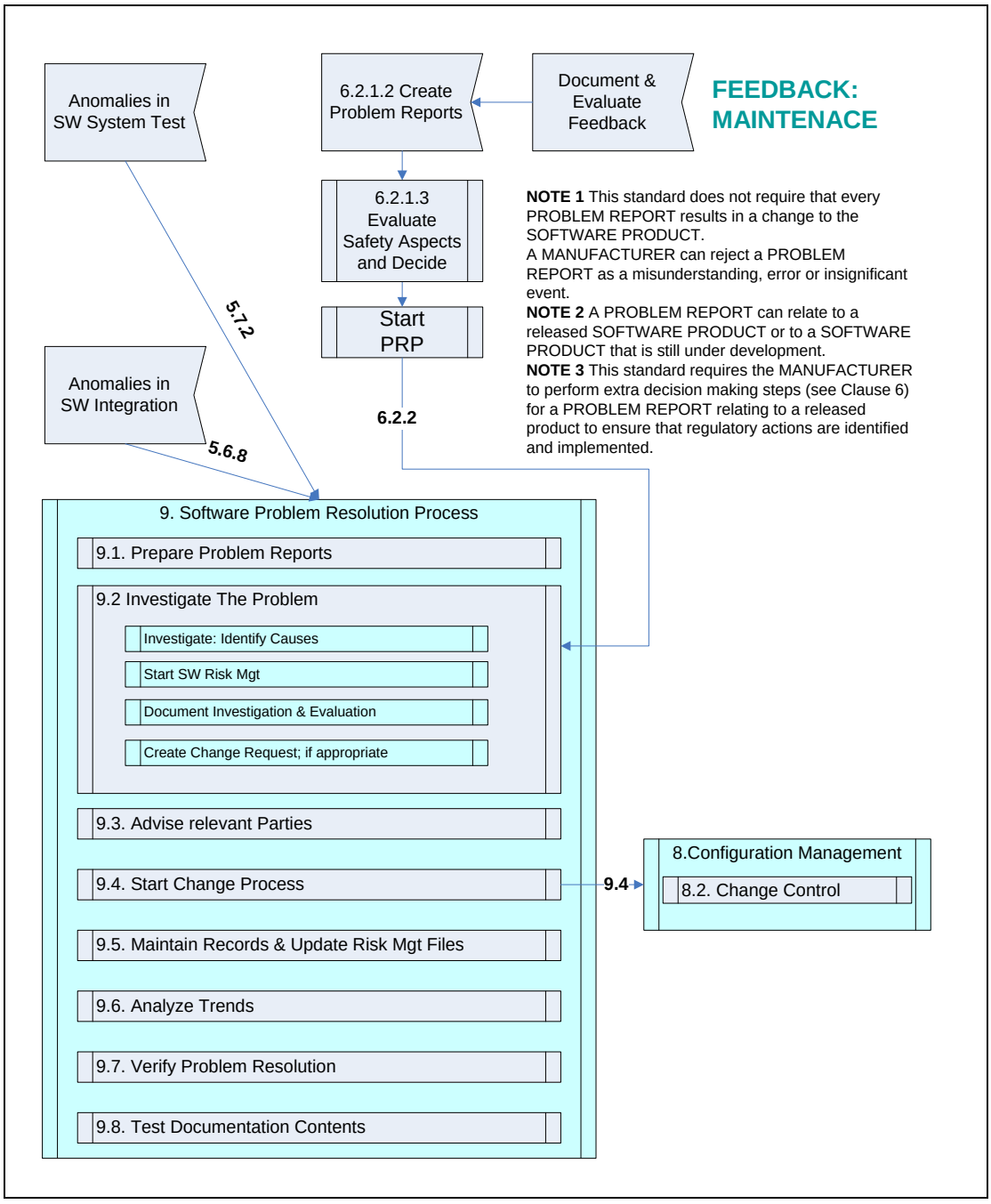


Figure 1 - Software Problem Resolution Process - Entry Points

<empty page>

Annex 2 SOUP selection, assessment & qualification

You may integrate a SOUP as a component in your product or integrate it as a product in your system. For simplicity sake the word 'product' is used in the text representing both component and integrated software. When a distinction is needed between product and system, this will be made clear in the text below (relates to questions 2.7.1 and 2.7.3).

The 'SOUP' flowchart (see Annex 2 Figure 2) provides an example of a process for the selection, assessment and qualification of SOUP suppliers. After searching (2) and identifying (3) a potential SOUP supplier that meets the specifications (4) you will designate the supplier as 'certified supplier'. The selection criteria may require a review if the supplier obtains the label 'critical'.

To determine whether the supplier is critical you must consider the potential impact of the SOUP on the safety (10) and efficacy (11) of your product. If the SOUP is intended for use as stand-alone software, i.e. as a product integrated into a system, you also need to determine who will take the manufacturer responsibility for the SOUP product (12). When answering these questions you should consider the changes you intend to make to the SOUP including any additional claims you may want to make with regards to the SOUP. Note that some companies may have additional criteria that could qualify a supplier as critical.

(10) If the SOUP has safety class B or C, then your supplier is automatically 'critical'.

(11) If you find that you do not have the capability to test yourself for the impact of the SOUP on your product's efficacy, then your supplier is 'critical'. E.g. an algorithm for the detection of clinical anomalies may involve costly or difficult clinical evaluations for which you want to rely on the evidence supplied by the manufacturer; similarly when you want to make changes to the SOUP or want to make new claims for which you rely on the supplier to provide the clinical evaluation.

(12) When you take manufacturer responsibility or act as authorized representative for a SOUP, also then your supplier is 'critical'.

If a supplier is designated as 'critical' you must perform an audit of the supplier. This can be an onsite audit or via a self-assessment questionnaire. Using your audit criteria you determine if a critical supplier can still be certified. You may, for example, accept the supplier if it has an established quality system (e.g. ISO 13485) or if it designs and tests its product according to your criteria. If the supplier does not meet your audit-criteria then the SOUP cannot be used. Note that audit criteria can be made dependent on the outcome of question 10 and 11. You may, for example, apply more stringent audit criteria if you rely on the manufacturer to test the effectiveness of the system, or you may request more stringent testing standards based on the safety impact of the SOUP.

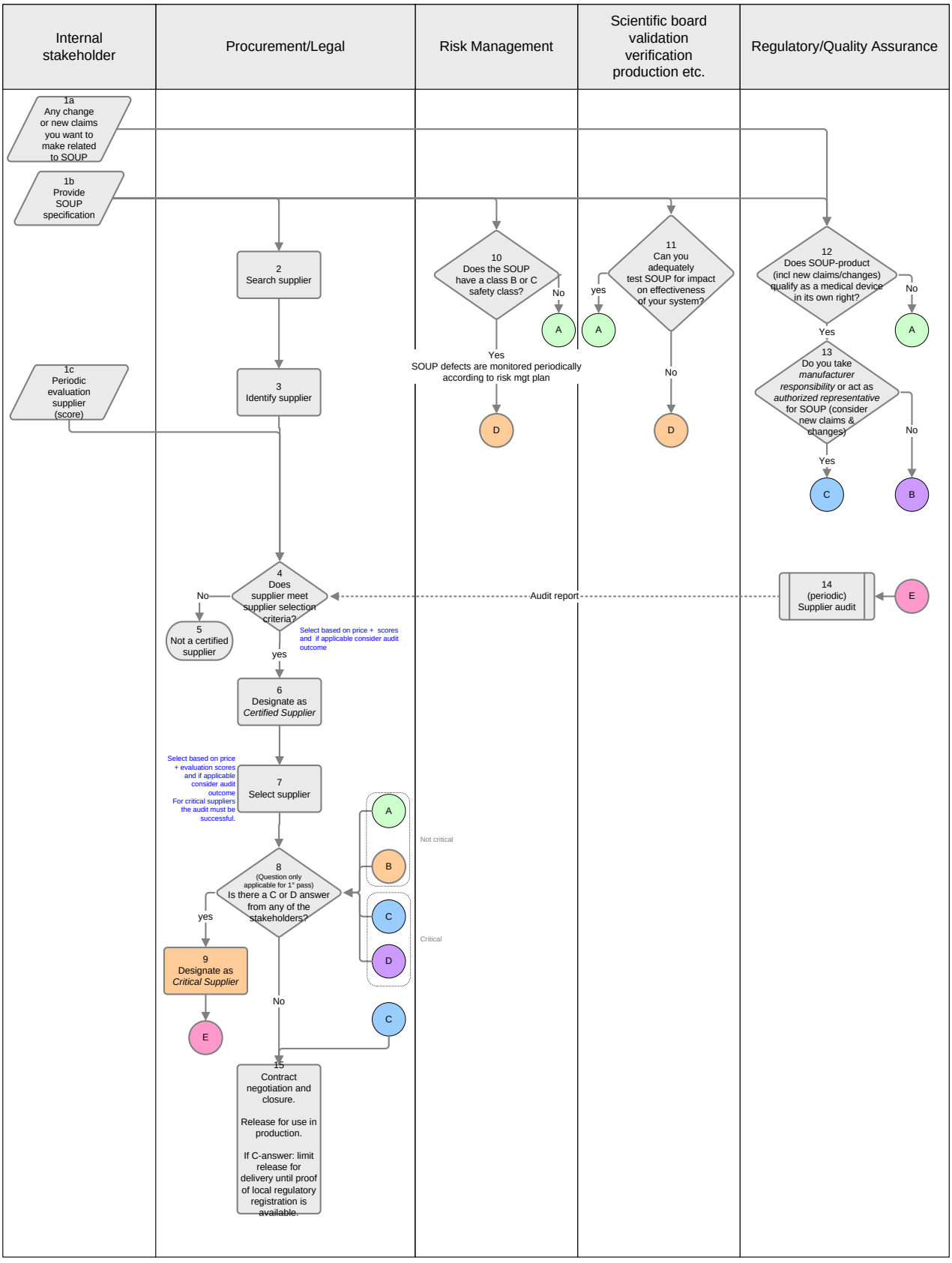


Figure 2 - SOUP selection, assessment & qualification

Annex 3 Traceability

Figure 3 shows an overview of the dependencies which need to be traced according to EN 62304 (relates to question 2.6.7).

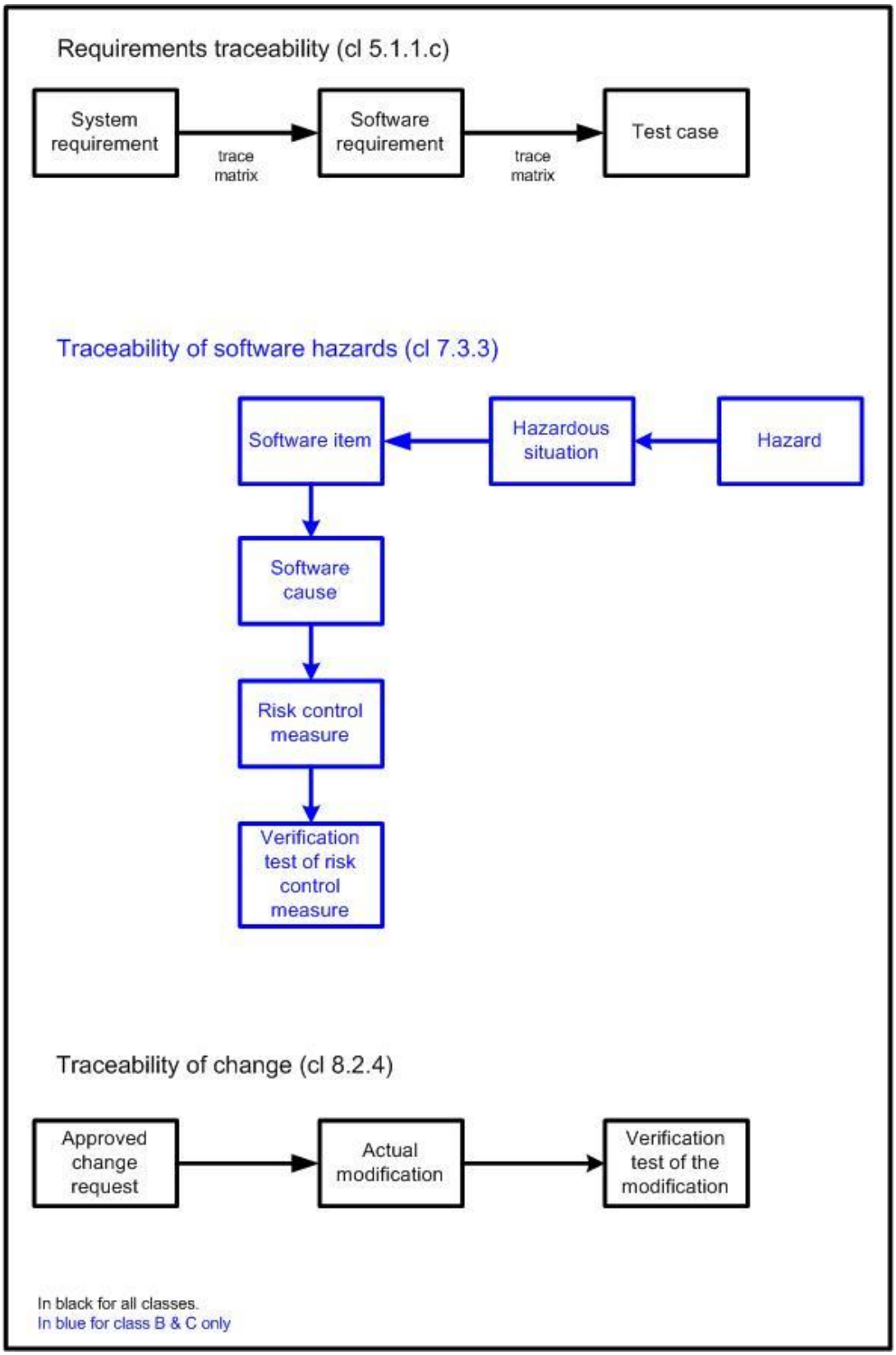


Figure 3 - Traceability

<empty page>

Annex 4 Position paper on direct diagnosis (COCIR, 2011)

(Relates to question 2.5.13)

Joint medical industry interpretation of the term "direct diagnosis"

The Directive 93/42/EEC concerning medical device is including an Annex IX ("Classification criteria") in which rule 10 point 3.2 uses the term "direct diagnosis".

As this term may be interpreted differently by the different stakeholders, COCIR and EUROM VI would like to share their own understanding of "direct diagnosis".

The paragraph is the following:

> *"Active devices intended for diagnosis are in Class IIa:*
> *- […]*
> *- if they are intended to allow **direct diagnosis** or monitoring of vital physiological processes, unless they are specifically intended for monitoring of vital physiological parameters, where the nature of variations is such that it could result in immediate danger to the patient, for instance variations in cardiac performance, respiration, activity of CNS in which case they are in Class IIb."*

COCIR and EUROM VI are of the opinion that the paragraph is to be interpreted as follows
If the active devices intended for diagnosis allow [a medical professional]:
1. to directly diagnose (e.g. diseases or conditions), or
2. to monitor vital physiological processes,

then in both case 1. and 2. the active devices are in class IIa, except when the active devices are intended to allow [a medical professional]:

3. to monitor vital physiological processes, where the nature of variations is such that it could result in immediate danger to the patient, for instance variations in cardiac performance, respiration, activity of CNS, in which case the active devices are in class IIb.

Definition of "a device allowing direct diagnosis"

A device is considered to allow direct diagnosis when it provides the diagnosis of the disease or condition by itself or when it provides decisive information for the diagnosis.

Rationale

1. Definition of "Active device for diagnosis".

An active device is defined in MDD Annex IX, point 1.6:

> *"Any active medical device, whether used alone or in combination with other medical devices, to supply information for detecting, diagnosing, monitoring or treating physiological conditions, states of health, illnesses or congenital deformities."*

This is to be interpreted as to supply information so a medical professional can use the information for the purposes mentioned (detecting, etc.).

2. Definition of "diagnosis".

"Diagnosis" is generally and commonly interpreted as

"the process of attempting to determine and/or identify a possible disease or condition and determine treatment. It includes follow-up of the progression of the disease, condition and treatment."

The diagnostic process begins by observing the patient for specific signs and symptoms and by taking a specific history, e.g. how did these signs and symptoms come about, etc. Specific signs, symptoms and historical clues allow the physician to perform a specific physical examination and order specific diagnostic investigations. The physician usually formulates a "short list" of likely diagnoses and requests further testing to confirm or rule-out competing diagnoses before providing treatment.

3. Definition of "direct" (as in the term "direct diagnosis").

Direct is to be interpreted with regard to completeness, i.e., without the necessity to acquire or take into account additional information. Then diagnosis can be made by a medical professional or by the medical device itself.

Note that the information available may not be absolutely complete (i.e. other parameters may be measured, the anamnesis may not be complete), but that the information is sufficient to imply a specific diagnosis. A device may provide information with varying medical relevance:

- *Indicative information*: the information can be used in a decision tree along with several other clinical and technical patient data for a healthcare professional to arrive at a diagnosis.
- *Decisive information: the information is one of the critical elements in determining the diagnosis.*

A device is considered to "allow direct diagnosis" when it provides the diagnosis of the disease or condition by itself or when it provides decisive information for the diagnosis. Indicative information is not sufficient to imply a 'direct diagnosis'.

4. Examples.

A non-exhaustive list of examples of devices that are used for direct diagnosis:

- Bone densitometers which classify the patient as "osteopenic" or "osteoporotic".
- ECG-systems which classify the patient as having "heart arrhythmia".
- Image processing applications which alter the image data in order to allow a medical professional to detect conditions, such as virtual colonoscopy for the detection of colonic polyps or vascular applications for detection of lung embolisms.

Acknowledgement

First of all, we thank everyone who responded to our call for questions. Without your contributions, this paper would not have been possible at all: the intent was to address specific questions, issues and matters from "real life" situations. Also, we thank all reviewers for their most valuable comments, and food for brainstorming and discussion.

Last but not least, Charles Rei deserves our gratitude because he, as native English speaker, definitively helped us to provide more clarity in our answers.

Frequently Asked Questions related to the Implementation of EN 62304:2006 with respect to MDD 93/42/EEC

Version 1.0

April 5, 2013