62A/1384/CD

COMMITTEE DRAFT (CD)

PROJECT NUMBER:	
IEC 80001-5-1 ED1	
DATE OF CIRCULATION:	CLOSING DATE FOR COMMENTS:
2020-01-31	**2020-04-24**
SUPERSEDES DOCUMENTS:	
62A/1360/CD,62A/1375A/CC	

IEC SC 62A : COMMON ASPECTS OF ELECTRICAL EQUIPMENT USED IN MEDICAL PRACTICE	
SECRETARIAT:	SECRETARY:
United States of America	Ms Hae Choe
OF INTEREST TO THE FOLLOWING COMMITTEES:	PROPOSED HORIZONTAL STANDARD:
TC 62,SC 62B,SC 62C,SC 62D	☐
	Other TC/SCs are requested to indicate their interest, if any, in this CD to the secretary.

FUNCTIONS CONCERNED:

☐ EMC ☐ ENVIRONMENT ☐ QUALITY ASSURANCE ☒ SAFETY

This document is still under study and subject to change. It should not be used for reference purposes.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

TITLE:

Safety, security and effectiveness in the implementation and use of connected medical devices or connected health software - Part 5-1: Security - Activities in the product lifecycle

NOTE FROM TC/SC OFFICERS:

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

Safety, security and effectiveness in the implementation and use of connected medical devices or connected health software

Part 5: Security

Part 5-1: Security - Activities in the product lifecycle

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 itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
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.

This Committee Draft of future International Standard IEC 80001-5-1 has been prepared by subcommittee 62A/ JWG7 of IEC technical committee 62 and ISO/TC 215/JWG 7.

Full information on the voting for the approval of this International Standard can be found in the report on voting indicated in the above table.
This document has been drafted in accordance with the ISO/IEC Directives, Part 2.
The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under "http://webstore.iec.ch" in the data related to the specific document. At this date, the document will be

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- withdrawn,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

The National Committees are requested to note that for this document the stability date is 2026
THIS TEXT IS INCLUDED FOR THE INFORMATION OF THE NATIONAL COMMITTEES AND WILL BE DELETED AT THE PUBLICATION STAGE.

Introduction

This International Standard specifies supplementary activities that the MANUFACTURER of Health Software – including software incorporated in MEDICAL DEVICES – shall perform towards the information SECURITY of the HEALTH SOFTWARE product / MEDICAL DEVICE.

> The project number and title of IEC 80001-5-1 are subject to an ongoing discussion on aligning numbers of several work items of JWG7. A new numbering scheme should systematically reflect the purpose and scope of respective work items. Please note that during the next draft stage, this project may have an updated number and title.

PROCESS requirements have been derived from IEC 62443-4-1 Product LIFECYCLE Management.

Implementations of these specifications will extend existing PROCESSES at the MANUFACTURER's organization –notably existing PROCESSES conforming to IEC 62304.

This document specifies activities for HEALTH SOFTWARE, the lifecycle of which can be part of an incorporating *product* project. Some activities specified in this document depend on input and support from the *product* LIFECYCLE (for example to define specific criteria). Examples include:

- RISK MANAGEMENT
- Requirements
- Testing
- Post-Market

In cases where activities for HEALTH SOFTWARE need support from PROCESSES at the product level, this document specifies respective requirements beyond the HEALTH SOFTWARE LIFECYCLE.

Similar to IEC 62304 this document does not prescribe a specific system of processes, but it requires that certain activities are being performed during the HEALTH SOFTWARE LIFECYCLE.

Chapter four specifies supporting activities in the HEALTH SOFTWARE LIFECYCLE.

Chapters five to eight specify activities and resulting output as part of the software LIFECYCLE PROCESS implemented by the MANUFACTURER. These specifications are arranged in the ordering of IEC 62304.

Chapter nine and ten specify activities and resulting output as part of the problem resolution PROCESS and quality management system respectively, implemented by the MANUFACTURER.

The scope of this document is limited to HEALTH SOFTWARE and its connectivity to its INTENDED ENVIRONMENT OF USE, based on IEC 62304, but with emphasis on information SECURITY.

This document is intended to supply minimum best practices towards secure software LIFECYCLE. Local legislation and regulation have to be considered.

This document requires establishing one or more PROCESSES that comprise of identified activities. The LIFECYCLE PROCESSES shall implement these activities. None of the requirements in this document requires to implement these activities as one single PROCESS or as separate PROCESSES.

Note: HEALTH SOFTWARE can be placed on the market as software or, incorporated into MEDICAL DEVICES or as software that in itself is considered a MEDICAL DEVICE, or incorporated into a general-purpose computing platforms.

Safety, security and effectiveness in the implementation and use of connected medical devices or connected health software

Part 5: Security

Part 5-1: Security - Activities in the product lifecycle

1 Scope

1.1 * Purpose

This document defines the secure LIFECYCLE requirements for development and MAINTENANCE of HEALTH SOFTWARE. The set of PROCESSES, activities, and tasks described in this document establishes a common framework for secure HEALTH SOFTWARE LIFECYCLE PROCESSES.

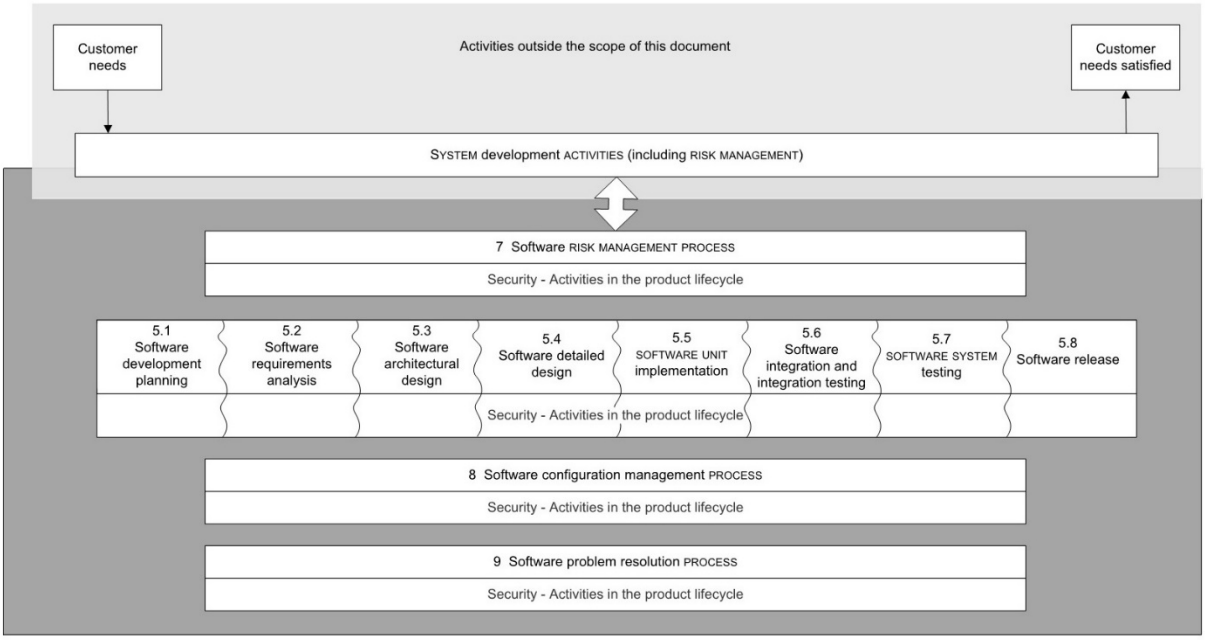


Fig. 1: HEALTH SOFTWARE LIFECYCLE Processes (derived from IEC 62304)

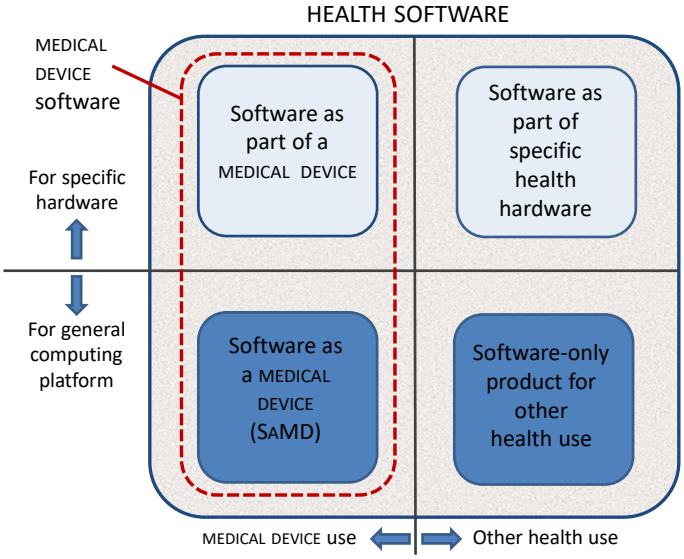
The purpose is to increase the information SECURITY of HEALTH SOFTWARE by establishing certain activities and tasks in the HEALTH SOFTWARE LIFECYCLE PROCESSES and also by increasing the SECURITY of SOFTWARE LIFECYCLE PROCESSES themselves.

This document excludes specification of ACCOMPANYING DOCUMENT contents.

1.2 * Field of application

This document applies to the development and MAINTENANCE of HEALTH SOFTWARE by a MANUFACTURER. MEDICAL DEVICE software is a subset of HEALTH SOFTWARE. Therefore, this document applies to:

– Software as part of a MEDICAL DEVICE;
– Software as part of specific health hardware;
– Software as a MEDICAL DEVICE (SaMD);
– Software-only product for other health use.



–

Fig. 2: HEALTH SOFTWARE field of application (source: IEC 62304 Ed 2)

2 Normative references

There are no normative references in this document.

3 Terms and definitions

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

Unless stated otherwise, definitions are aligned with ISO 81001-1.

3.1

ACCOMPANYING DOCUMENT

document accompanying a HEALTH SOFTWARE and HEALTH IT SYSTEM or an accessory, containing information for the responsible organization or operator, particularly regarding SAFETY

[SOURCE: ISO 81001-1:202x, 3.1]

3.2

ARCHITECTURE

organizational structure of a SYSTEM or component

[SOURCE: IEEE Std 24765-2010, 3.150, definition 2]

3.3

ASSET

physical or digital entity that has value to an individual, an organization or a government

[SOURCE: ISO 81001-1:202x, 3.3]

3.4

ATTACK

attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of an ASSET

[SOURCE: ISO/IEC 27000:2016, 2.3]

3.5

AUDIT LOG

Chronological sequence of audit records, each of which contains data about a specific event..

Note to entry: Audit logs are used to protect against claims that repudiate responsibility for an action.

[SOURCE: ISO/IEC 27789:2013, modified – added note to entry]

3.6

AUDIT RECORD

record of the resources which were accessed and/or used by whom

3.7

AVAILABILITY

property of being accessible and usable upon demand by an authorized entity

[SOURCE: ISO/IEC 27000:2016, 2.9]

3.8

CONFIDENTIALITY

property that information is not made available or disclosed to unauthorized individuals, entities, or PROCESSES

[SOURCE: ISO/IEC 24767-1:2008, 2.1.2]

3.9

EXPLOIT (noun)

defined way to breach the SECURITY of information systems through some VULNERABILITY

[SOURCE: ISO/IEC 27039:2015]

3.10

HEALTH IT INFRASTRUCTURE

combined set of IT ASSETS available to the individual or organization for developing, configuring, integrating, maintaining, and using IT services and supporting health, patient care and other organizational objectives.

Note 1 to entry: As per the definition for ASSET this can include the following:

a) data and information;

b) HEALTH SOFTWARE (including software in/as a MEDICAL DEVICES, health applications, middleware, and operating system software);

c) hardware components such as computers, mobile devices, servers, databases, and networks;

d) services, including SECURITY, software development, IT operations and externally provided services such as data centres, internet and software-as-a-service and cloud solutions;

e) people, and their qualifications, skills and experience;

f) technical procedures and documentation to manage and support the HEALTH IT INFRASTRUCTURE;

g) HEALTH IT SYSTEMS that are configured and implemented to address organizational objectives by leveraging the ASSETS;

h) intangibles, such as reputation and image.

[SOURCE: ISO 81001-1:202x, 3.21]

3.11

HEALTH IT SYSTEM

a combination of interacting health information elements (including HEALTH SOFTWARE, MEDICAL DEVICES, IT hardware, interfaces, data, procedures and documentation) that is configured and implemented to support and enable an individual or organization's specific health objectives.

[SOURCE: ISO 81001-1:202x, 3.22]

3.12

HEALTH SOFTWARE

software intended to be used specifically for managing, maintaining, or improving health of individual persons, or the delivery of care, or which has been developed for the purpose of being incorporated into a MEDICAL DEVICE.

Note 1 to entry: HEALTH SOFTWARE fully includes what is considered software as a MEDICAL DEVICE.

[SOURCE: ISO 81001-1:202x, 3.23]

3.13

HEALTHCARE DELIVERY ORGANIZATION

HDO

facility or enterprise such as a clinic or hospital that provides healthcare services

[SOURCE: ISO 81001-1:202x, 3.24]

356

357 **3.14**

358 **INTEGRITY**

359 property of accuracy and completeness

360 [SOURCE: ISO/IEC 27000:2016, 2.40]

361

362 **3.15**

363 **INTENDED ENVIRONMENT OF USE**

364 conditions and setting in which users interact with the HEALTH SOFTWARE – as specified by the
365 MANUFACTURER

366

367 **3.16**

368 **INTENDED USE**

369 **INTENDED PURPOSE**

370 use for which a PRODUCT, PROCESS or service is intended according to the specifications,
371 instructions and information provided by the MANUFACTURER

372 Note 1 to entry: The intended medical indication, patient population, part of the body or type of
373 tissue interacted with, user profile, INTENDED ENVIRONMENT OF USE, and operating principle are
374 typical elements of the intended use.

375 [SOURCE: ISO 81001-1:202x, 3.28, note 1 to entry modified – “use environment” replaced by
376 “INTENDED ENVIRONMENT OF USE”.]

377

378 **3.17**

379 **LEGACY SOFTWARE**

380 HEALTH SOFTWARE placed on the market before the publication of this document for which the
381 MANUFACTURER seeks conformance retrospectively.

382

383 **3.18**

384 **LIFE CYCLE**

385 series of all phases in the life of a product or system, from the initial conception to final
386 decommissioning and disposal

387 [SOURCE: ISO 81001-1:202x, 3.32]

388

389 **3.19**

390 **MANUFACTURER**

391 natural or legal person with responsibility for the design and/or manufacture of HEALTH
392 SOFTWARE, MEDICAL DEVICES or HEALTH IT SYSTEMS with the intention of making them available
393 for use, under his name, whether or not such a HEALTH SOFTWARE, MEDICAL DEVICE or HEALTH
394 IT SYSTEMS is designed and/or manufactured by that person himself or on his behalf by another
395 person(s)

396 Note 1 to entry: The natural or legal person has ultimate legal responsibility for ensuring
397 compliance with all applicable regulatory requirements for the MEDICAL DEVICE in the countries
398 or jurisdictions where it is intended to be made available or sold, unless this responsibility is
399 specifically imposed on another person by the Regulatory Authority (RA) within that jurisdiction.

Note 2 to entry: The MANUFACTURER's responsibilities are described in other GHTF guidance documents. These responsibilities include meeting both pre-market requirements and post-market requirements, such as adverse event reporting and notification of corrective actions.

Note 3 to entry: "Design and/or manufacture" may include specification development, production, fabrication, assembly, processing, packaging, repackaging, labelling, relabelling, sterilization, installation, or remanufacturing of HEALTH SOFTWARE, a HEALTH IT SYSTEMS or a MEDICAL DEVICE; or putting a collection of devices, and possibly other products, together for a medical purpose.

Note 4 to entry: Any person who assembles or adapts a MEDICAL DEVICE that has already been supplied by another person for an individual patient, in accordance with the instructions for use, is not the MANUFACTURER, provided the assembly or adaptation does not change the intended use of the HEALTH SOFTWARE, a HEALTH IT SYSTEM or MEDICAL DEVICE.

Note 5 to entry: Any person who changes the intended use of, or modifies, HEALTH SOFTWARE, a HEALTH IT SYSTEM or a MEDICAL DEVICE without acting on behalf of the original MANUFACTURER and who makes it available for use under his own name, should be considered the MANUFACTURER of the modified MEDICAL DEVICE.

Note 6 to entry: An authorised representative, distributor or importer who only adds its own address and contact details to the HEALTH SOFTWARE, a HEALTH IT SYSTEM or MEDICAL DEVICE or the packaging, without covering or changing the existing labelling, is not considered a MANUFACTURER.

Note 7 to entry: To the extent that an accessory is subject to the regulatory requirements of HEALTH SOFTWARE, a HEALTH IT SYSTEM or a MEDICAL DEVICE, the person responsible for the design and/or manufacture of that accessory is considered to be a MANUFACTURER.

Note 8 to entry: Some differences can occur in the definitions in regulations of each country.

[SOURCE: ISO 81001-1:2020, 3.33, modified – added note 8 to entry, extended to HEALTH SOFTWARE]

3.20

MEDICAL DEVICE

instrument, apparatus, implement, machine, appliance, implant, reagent for in vitro use, software, material or other similar or related article, intended by the MANUFACTURER to be used, alone or in combination, for human beings, for one of more of the specific medical 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,

— cleaning, disinfection, or sterilization of MEDICAL DEVICES,

— providing information by means of in vitro examination of specimens derived from the human body,

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

Note 1 to entry:PRODUCTS which could be considered to be MEDICAL DEVICES in some jurisdictions but not in others include:

— disinfection substances,

— aids for persons with disabilities,

— devices incorporating animal and/or human tissues,

— devices for in-vitro fertilization or assisted reproductive technologies.

[SOURCE: ISO 81001-1:202x, 3.34]

3.21

PROCESS

set of interrelated or interacting activities that use inputs to deliver an intended result (outcome)

[SOURCE: ISO 81001-1:202x, 3.38, modified – added “(outcome)” after “result”.]

3.22

PRODUCT

output of an organization that can be produced without any transaction taking place between the organization and the customer

Note 1 to entry: Production of a PRODUCT is achieved without any transaction necessarily taking place between provider and customer, but can often involve this service element upon its delivery to the customer.

Note 2 to entry: The dominant element of a PRODUCT is that it is generally tangible.

[SOURCE: ISO 81001-1:202x, 3.39]

3.23

RESIDUAL RISK

risk remaining after RISK CONTROL measures have been implemented

[SOURCE: ISO 81001-1:202x, 3.42]

3.24

RISK CONTROL

PROCESS in which decisions are made and measures implemented by which risks are reduced to, or maintained within, specified levels

[SOURCE: ISO 81001-1:202x, 3.47]

3.25

RISK MANAGEMENT

systematic application of management policies, procedures and practices to the tasks of analysing, evaluating, controlling and monitoring risk

[SOURCE: ISO 81001-1:202x, 3.50]

3.26

SAFETY

freedom from unacceptable risk

Note 1 to entry: Risk is the combination of probability of harm and severity of harm (see ISO/IEC guide 51:2014).

Note 2 to entry: SECURITY incidents can lead to harm and can therefore have an impact on SAFETY.

[SOURCE: ISO 81001-1:202x, 3.55, modified – added notes to entry.]

3.27

SECURITY

CYBERSECURITY

A state where information and systems are protected from unauthorized activities, such as access, use, disclosure, disruption, modification, or destruction to a degree that the related risks to CONFIDENTIALITY, INTEGRITY, and AVAILABILITY are maintained at an acceptable level throughout the LIFE,CYCLE.

[SOURCE: ISO 81001-1:202x, 3.56]

3.28

SECURITY CAPABILITY

broad category of technical, administrative or organizational controls to manage risks to CONFIDENTIALITY, INTEGRITY, AVAILABILITY and accountability of data and systems

[SOURCE: ISO 81001-1:202x, 3.57]

3.29

SOFTWARE COMPOSITION ANALYSIS

(electronic) analysis of binaries.

Note to entry: SOFTWARE COMPOSITION ANALYSIS can be supported by tools or online services.

3.30

SOFTWARE ITEM

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

[SOURCE: IEC 62304:2015]

3.31

SOFTWARE MAINTENANCE

modification of **HEALTH SOFTWARE** after release for **INTENDED USE**, for one or more of the following reasons:

a) corrective, as fixing faults;

b) adaptive, as adapting to new hardware or software platform;

c) perfective, as implementing new requirements;

d) preventive, as making the product more maintainable.

Note 1 to entry: See also ISO/IEC 14764:2006, 3.10.

[SOURCE: IEC 82304-1:2016, 3.21, modified – In the definition, the words "HEALTH SOFTWARE PRODUCT" have been replaced by "HEALTH SOFTWARE", and reference 3.10 has been added to the note to entry; and “hard-“ has been replaced by “hardware”]

3.32

THREAT

potential for violation of SECURITY, which exists when there is a circumstance, capability, action, or event that could breach SECURITY and cause harm

[SOURCE: ISO 81001-1:202x, 3.62]

3.33

THREAT MODEL

documented result of THREAT MODELLING

3.34

THREAT MODELLING

Threat Modelling: systematic exploration technique to expose any circumstance or event having the potential to cause damage to a system in the form of destruction, disclosure, modification of data, or denial of service.

[SOURCE. ISO 24765:2017, modified – replaced “harm” with “damage”]

3.35

USE ENVIRONMENT

actual conditions and setting in which users interact with the HEALTH SOFTWARE

[SOURCE. IEC 62366-1:2015; 3.20, modified]

Note to entry: For the purpose of this document, that includes data interfaces.

3.36

VALIDATION

confirmation, through the provision of objective evidence, that the requirements for a specific INTENDED USE or application have been fulfilled.

Note 1 to entry: The objective evidence needed for a VALIDATION is the result of a test or other form of determination such as performing alternative calculations or reviewing documents.

Note 2 to entry: The word “validated” is used to designate the corresponding status.

Note 3 to entry: The use conditions for VALIDATION can be real or simulated.

[SOURCE: ISO 9000:2015, 3.8.13]

3.37

VERIFICATION

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

Note 1 to entry: The objective evidence needed for a VERIFICATION can be the result of an inspection or of other forms of determination such as performing alternative calculations or reviewing documents.

Note 2 to entry: The activities carried out for VERIFICATION are sometimes called a qualification PROCESS.

Note 3 to entry: The word "verified" is used to designate the corresponding status.

[SOURCE: ISO 81001-1:20xx, 3.66, modified – Note 1 to entry has been rephrased.]

3.38

VULNERABILITY

flaw or WEAKNESS in a system's design, implementation, or operation and management that could be exploited to violate the system's SECURITY policy

[SOURCE: ISO 81001-1:202x, 3.67]

3.39

WEAKNESS

kind of deficiency.

Note: A WEAKNESS can result in a SECURITY risk.

[SOURCE: ISO 81001-1:202x, 3.68, modified]

4 General Requirements

4.1 Quality Management

4.1.1 Documentation of PROCESS

The MANUFACTURER shall establish and maintain a documented quality management system that is the basis of SECURITY activities in the PRODUCT LIFECYCLE.

Throughout this document “establish an activity(s) X“ means that the MANUFACTURER shall document X and shall ensure that X is done properly.

Note 1: This quality management system can be implemented according to ISO 13485 or other equivalent QMS.

Note 2: SECURITY considerations in quality management systems are described in Clause 10.

4.1.2 Identification of responsibilities

The MANUFACTURER shall establish ACTIVITY(S) to identify the organizational roles and personnel responsible for each of the ACTIVITIES and PROCESSES required by this document.

Note 1: Personnel can be identified through functional roles instead of names.

4.1.3 Identification of applicability

The MANUFACTURER shall identify the PRODUCTS or parts of PRODUCTS to which this standard applies.

Note: This requirement is not about product instances (and their identification) but about types of products or their parts. Having this activity(s) means that the MANUFACTURER has criteria for identifying which of its products are to be developed, maintained and supported using the activities required by this document.

4.2 THREAT / RISK MANAGEMENT

The MANUFACTURER shall establish a PROCESS for managing RISKS associated with SECURITY. This PROCESS shall provide methods for identifying VULNERABILITIES, estimating and evaluating the associated THREATS, controlling these THREATS, and monitoring the effectiveness of the RISK CONTROL (SECURITY) measures, taking into account the INTENDED USE and the INTENDED ENVIRONMENT OF USE of the HEALTH SOFTWARE.

The MANUFACTURER shall establish the criteria for RISK acceptability that shall be applied when determining the appropriate way to address each issue.

The SECURITY RISK MANAGEMENT should incorporate outcomes of the THREAT Modelling Activity(s) and other inputs such as guidelines and state-of-the-art.

Detailed PROCESS steps are described in Clause 7.

Note: This PROCESS can be part of an existing general RISK MANAGEMENT PROCESS: SECURITY RISK MANAGEMENT can be conducted under the framework of ISO 14971 with an appropriate mapping of VULNERABILITY, THREAT and other SECURITY related terms. (See ISO/TR 24971:20XX for possible mapping.)

The MANUFACTURER shall document any RESIDUAL RISK associated with a VULNERABILITY that remains in the system.

See Annex D on THREAT MODELLING

5 Software development PROCESS

5.1 Software Development Planning

5.1.1 ACTIVITIES in the lifecycle PROCESS

The MANUFACTURER shall establish general lifecycle ACTIVITIES – from conception to decommissioning - that are consistent and integrated with a commonly accepted PRODUCT development PROCESS including but not limited to:

a) configuration management with change controls and change history;

b) PRODUCT description and requirements definition with requirements TRACEABILITY;

Note 1: TRACEABILITY links requirements to their origin and traces them throughout the project life cycle to design elements, implementation and test cases (ISO 24765:2017, modified)

c) software or hardware design and implementation practices, such as modular design;

d) repeatable testing VERIFICATION and VALIDATION PROCESS;

e) review and approval of all development PROCESS records; and

f) Product support.

Note 2: Product Support means providing of information, assistance and training to install and make Health Software operational in its intended environment and to distribute improved capabilities to users (ISO 24765:2017)

Note 3: The underlying practice SM-5 in IEC 62443-4-1 specifies a requirement for PROCESS tailoring. This is meant to base an appropriate reduction of security activities on the outcomes of the security analysis.

5.1.2 Development environment SECURITY

The MANUFACTURER shall establish procedural and technical controls for protecting the IT infrastructure used for development, production delivery and maintenance from unauthorized access, corruption and deletion. This includes protecting the HEALTH SOFTWARE during design, implementation, updates, testing and release.

5.1.3 Secure coding standards

The MANUFACTURER shall establish and maintain secure coding standards. (See Annex A, Rationale)

5.2 HEALTH SOFTWARE Requirements Analysis

5.2.1 HEALTH SOFTWARE SECURITY requirements

The MANUFACTURER shall establish an ACTIVITY(S) for ensuring that SECURITY requirements are documented for the HEALTH SOFTWARE including requirements for SECURITY capabilities related to installation, operation, MAINTENANCE and decommissioning.

The MANUFACTURER shall establish an ACTIVITY(S) ensuring that SECURITY requirements include the following information:

a) The scope and boundaries of the components of the HEALTH SOFTWARE, in general terms in both a physical and logical way;

b) Information on interfaces: The integration capabilities of the HEALTH SOFTWARE'S Identity and Access Management with that of the HDO (where applicable); AND the integration capabilities of the HEALTH SOFTWARE within the operating environment of the HDO

c) Controls implemented in the product

d) Design for (SECURITY) update of HEALTH SOFTWARE including the update of incorporated software from external sources.

Note 1: IEC TR 60601-4-5 gives guidance on the specification of SECURITY capabilities and their documentation in the accompanying documents and provides a method of determining requirements from the SECURITY CAPABILITY level.

Note 2: IEC TR 80001-2-2 specifies SECURITY -related needs, risks and controls as a guidance for disclosure and communication between the MANUFACTURER and the HEALTHCARE DELIVERY ORGANIZATION.

Note 3: The PRODUCT requirements PROCESS interfaces with HEALTH SOFTWARE requirements. Some technical controls can be implemented at PRODUCT level. (for example by hardware).

5.2.2 SECURITY requirements review

The MANUFACTURER shall establish an ACTIVITY(S) which incorporates reviews to ensure that SECURITY requirements are reviewed, updated as necessary and approved to ensure clarity, validity, alignment with the SECURITY RISK MANAGEMENT PROCESS, and their ability to be verified.

The MANUFACTURER shall document the level of independence of the reviewers. Each of the following representative disciplines shall participate in this ACTIVITY(S):

a) architects/developers (those who will implement the requirements);
b) testers (those who will validate that the requirements have been met);
c) cross-functional experts (may include those with clinical expertise); and
d) SECURITY advisor.

Note 1: A single person can be responsible for multiple disciplines. It is not advisable to have a single person representing all disciplines.

Note 2: The list of disciplines has to be documented at least once per project.

Note 3: QMS like that of ISO 13485 require consideration of independence of reviewers.

5.2.3 SECURITY requirements for externally provided components

The MANUFACTURER shall establish an ACTIVITY(S) that identifies and manages the SECURITY risks of all externally provided components intended to be used with the HEALTH SOFTWARE.

Note: This ACTIVITY(S) can be part of supply chain security ACTIVITIES.

5.3 Software Architectural Design

5.3.1 Defense in depth ARCHITECTURE/design

The MANUFACTURER shall establish an ACTIVITY(S) to specify a secure ARCHITECTURE.

At each stage of development, the MANUFACTURER shall consider DEFENSE-IN-DEPTH and assign technical responsibilities to each layer of defense.

Note: DEFENSE-IN-DEPTH may include security requirements to be transferred to and implemented by the HDO.

5.3.2 Document secure design best practices

The MANUFACTURER shall establish an ACTIVITY(S) to identify, enforce and maintain secure design practices. The MANUFACTURER shall document secure design best practices, which should include but are not limited to:

a) least privilege (granting only the privileges to users/software necessary to perform intended operations);
b) using proven secure components/designs where possible;
c) economy of mechanism (striving for simple designs);

d) using secure design patterns;

e) ATTACK SURFACE reduction;

f) documenting all trust boundaries as part of the design; and

g) removing debug ports, headers and traces from circuit boards used during development from production hardware or documenting their presence and the need to protect them from unauthorized access;

h) removing debug code (variables, break points, console logging) used during development from production software or documenting the presence of debug code and the need to protect it from unauthorized access.

Note: See Annex B: “Guidance“

5.3.3 SECURITY architectural design review

The MANUFACTURER shall implement an architectural review of the HEALTH SOFTWARE with respect to (behavior under adverse conditions)

a) effective segregation of SOFTWARE ITEMS,

b) the secure design best practices (see 5.3.2)

c) potential SECURITY flaws introduced by the architecture.

Note: Segregation uses technical controls in design and implementation in order to ensure that SOFTWARE ITEMS cannot be influenced by other SOFTWARE ITEMS of the HEALTH SOFTWARE in an unintended way.

5.4 Software Detailed Design

5.4.1 Secure detailed design best practices

The MANUFACTURER shall establish and maintain the use of best practices towards a secure design, taking into account

a) software technology at application level (like algorithms, methods …)

b) the programming technology used, (e.g. programming language)

c) the secure design best practices in 5.3.2.

5.4.2 Secure HEALTH SOFTWARE design

The MANUFACTURER shall establish an ACTIVITY(S) to develop and document a secure design that identifies and characterizes each interface of the HEALTH SOFTWARE including physical and logical interfaces. As appropriate, the design shall identify:

a) an indication of whether the interface is externally accessible (by other PRODUCTS) or internally accessible - between components of the HEALTH SOFTWARE- or both

b) SECURITY implications of the HEALTH SOFTWARE SECURITY context on the external interface

c) potential users of the interface and the ASSETS that can be accessed through the interfaces (directly or indirectly)

d) a determination of whether the static design accesses interfaces across trust boundaries;

e) SECURITY considerations, assumptions and/or constraints associated with the use of the interface within the HEALTH SOFTWARE SECURITY context; including applicable THREATS,

f) the SECURITY roles, privileges/rights and access control permissions needed to use the interface and to access the ASSETS defined in c)

g) the SECURITY capabilities and/or compensating mechanisms used to safeguard the interface and the ASSETS identified in c) including run-time validation of inputs as well as handling outputs and errors;

h) the use of third-party components to implement the interface and their SECURITY capabilities

i) documentation that describes how to use the interface if it is externally accessible; and

j) description of how the design mitigates the THREATS identified in the THREAT model.

Note: The SECURITY context for HEALTH SOFTWARE is derived from the INTENDED ENVIRONMENT OF USE at PRODUCT-level, considering also the configuration and integration of HEALTH SOFTWARE.

5.4.3 Detailed design VERIFICATION towards SECURITY

The MANUFACTURER shall establish an ACTIVITY(S) for conducting design reviews to identify, characterize and track to closure WEAKNESSES associated with each significant revision of the secure design including but not limited to:

a) SECURITY requirements that were not adequately addressed by the design
b) THREATS and their ability to exploit VULNERABILITIES in PRODUCT interfaces, trust boundaries and ASSETS
c) Identification, documentation and characterization of (detailed) design best-practices that were not followed (5.3.2 and 5.4.1)

5.5 Software Unit Implementation and VERIFICATION

5.5.1 Secure Coding Standards

The MANUFACTURER shall establish an implementation ACTIVITY(S) following secure coding standards.

5.5.2 SECURITY implementation review

The MANUFACTURER shall establish an ACTIVITY(S) to ensure that implementation reviews are performed for identifying, characterizing and feed into the Problem Resolution Process all SECURITY-related issues associated with the implementation of the secure design including:

a) identification of SECURITY requirements (see 5.2) that were not adequately addressed by the implementation;

   Note: Requirements allocation, including SECURITY requirements, is part of typical design PROCESSES.

b) identification of secure coding standards that were not followed (for example, use of banned functions or failure to apply the principle of least privilege);

c) Static Code Analysis (SCA) for source code to determine secure coding errors using the secure coding standard for the supported programming language, as established in 5.1.3

d) review of the implementation and its traceability to the SECURITY capabilities defined to support the SECURITY design (see 5.3 and 5.4); and

e) examination of THREATS and their ability to exploit implementation interfaces, trust boundaries and ASSETS (see 5.3 and 5.4).

5.6 Software integration testing

The MANUFACTURER may perform some of the Software System Testing as a part of Software Integration Testing (see 5.7)

5.7 Software System Testing

5.7.1 SECURITY requirements testing

The MANUFACTURER shall establish an ACTIVITY(S) for verifying that the HEALTH SOFTWARE SECURITY functions meet the SECURITY requirements and that the HEALTH SOFTWARE handles error scenarios and invalid input. Based on the INTENDED ENVIRONMENT OF USE, types of testing shall include:

a) Functional testing of SECURITY requirements;

b) Performance and scalability testing; and

c) Boundary/edge condition, stress and malformed or unexpected input tests with potential SECURITY consequences;

Note: See Annex B 7.3

5.7.2 THREAT mitigation testing

The MANUFACTURER shall establish an ACTIVITY(S) for testing the effectiveness of the mitigation for the THREATS identified and validated in the THREAT model. Activities shall include:

a) creating and executing plans to ensure that each mitigation implemented to address a specific THREAT has been adequately tested to ensure that the mitigation works as designed; and

b) creating and executing plans for attempting to thwart each mitigation.

5.7.3 VULNERABILITY testing

The MANUFACTURER shall establish an ACTIVITY(S) for performing tests that focus on identifying and characterizing potential SECURITY VULNERABILITIES in the HEALTH SOFTWARE. Known VULNERABILITY testing shall be based upon, at a minimum, recent contents of an established, industry-recognized, public source for known VULNERABILITIES. As appropriate, testing shall include:

a) abuse case or malformed or unexpected input testing focused on uncovering SECURITY issues. This shall include manual or automated abuse case testing and specialized types of abuse case testing on all external interfaces and protocols. Examples include fuzz testing and network traffic load testing and capacity testing;

b) ATTACK surface analysis to determine all avenues of ingress and egress to and from the system, common VULNERABILITIES including but not limited to weak access-control-lists (ACLs), exposed ports and services running with elevated privileges;

c) black box known VULNERABILITY scanning focused on detecting known VULNERABILITIES in (if applicable) hardware, host, interfaces or software components

Note: For example, this could be a network based known VULNERABILITY scan;

d) for compiled software, SOFTWARE COMPOSITION ANALYSIS on all binary executable files, including embedded firmware, delivered by a component supplier to be used with HEALTH SOFTWARE. This analysis shall be used to support testing ACTIVITIES in minimum detection of:

1) known VULNERABILITIES in the HEALTH SOFTWARE components;
2) linking to vulnerable libraries;
3) SECURITY rule violations; and
4) compiler settings that can lead to VULNERABILITIES;

e) dynamic runtime resource management testing that detects flaws not visible under static code analysis, including but not limited to denial of service conditions due to failing to release runtime handles, memory leaks and accesses made to shared memory without authentication. This testing shall be applied if such tools are available

Note: Dynamic runtime testing cannot be done effectively without tools.

5.7.4 Penetration testing

The MANUFACTURER shall establish an ACTIVITY(S) to identify and characterize WEAKNESSES via tests that focus on discovering and exploiting SECURITY VULNERABILITIES in the HEALTH SOFTWARE.

5.8 Software Release

5.8.1 Resolve findings prior to release

The MANUFACTURER shall establish an ACTIVITY(S) to ensure that all findings from system testing have been handled by the Problem Resolution PROCESS (Clause 9).

889 ### 5.8.2 Release documentation

890 As a part of the software release ACTIVITY(s) the MANUFACTURER shall establish requirements
891 for ACCOMPANYING DOCUMENTS:

892 a) Secure operation guidelines

893 b) Account management guidelines (if applicable)

894

895 ### 5.8.3 File INTEGRITY

896 The MANUFACTURER shall establish an ACTIVITY(S) to provide an INTEGRITY VERIFICATION
897 mechanism for all scripts, executables and other security-relevant files used with a HEALTH
898 SOFTWARE.

899

900 ### 5.8.4 Controls for private keys

901 The MANUFACTURER shall have procedural and technical controls in place to protect private keys
902 used for code signing from unauthorized access or modification.

903

904 Note: This refers to the software supply chain and the focus is on code signing to support secure
905 distribution and delivery of HEALTH SOFTWARE.

906

907 ### 5.8.5 Assessing and addressing SECURITY-related issues

908 The MANUFACTURER shall establish an ACTIVITY(S) for verifying that a HEALTH SOFTWARE or an
909 update is not released until its SECURITY-related issues have been addressed and tracked to
910 closure (see 9.5). This includes issues associated with:

911 a) requirements (see 5.2);

912 b) security by design (see 5.3 and 5.4);

913 c) implementation (see 5.5);

914 d) VERIFICATION / VALIDATION (see 5.5); and

915 e) SECURITY defect management (see 9.4).

916

917 ### 5.8.6 ACTIVITY Completion

918 The MANUFACTURER shall establish an ACTIVITY(S) for verifying that, prior to HEALTH SOFTWARE
919 release, all applicable SECURITY-related PROCESSES required by this standard have been
920 completed with records documenting the completion of each ACTIVITY(S) or PROCESS.

6 SOFTWARE MAINTENANCE PROCESS

6.1 SECURITY Update Management

6.1.1 SECURITY Update VERIFICATION

The MANUFACTURER shall establish an ACTIVITY(S) for verifying that SECURITY updates created by the MANUFACTURER address the intended SECURITY VULNERABILITIES

The MANUFACTURER shall establish an ACTIVITY(S) for verifying that SECURITY updates do not introduce unintended effects to functional or quality attributes of Health Software. Such security updates include but are not limited to updates created by:

a) the HEALTH SOFTWARE MANUFACTURER;
b) suppliers of components used in the HEALTH SOFTWARE; and
c) suppliers of components or platforms on which the HEALTH SOFTWARE depends.

The manufacturer may define that for certain components or platforms, there is a shared responsibility for such VERIFICATION.

Note: Also see Clause 9 “Problem Resolution PROCESS”

6.1.2 Modification Implementation

The MANUFACTURER shall establish an ACTIVITY(S) to ensure that each applicable update for supported PRODUCTS and PRODUCT versions is made available to PRODUCT users in a manner that facilitates INTEGRITY VERIFICATION of the SECURITY update.

6.2 Post-Market activities for HEALTH SOFTWARE

6.2.1 SECURITY update documentation

The MANUFACTURER shall establish an ACTIVITY(S) to ensure that documentation about HEALTH SOFTWARE SECURITY updates is made available that includes but is not limited to:

a) risks (potential impact to SAFETY, effectiveness, SECURITY) of not applying the update and mitigations that can be used for updates that are not approved or deployed by the ASSET owner;
b) the potential for patient harm – if the update is not being installed;
c) a strategy to reduce the risk of patient harm to an acceptable and controlled level.

6.2.2 Dependent component

The MANUFACTURER shall establish a policy to provide dependent component or operating system SECURITY updates to PRODUCT users and ensure that this policy is followed. At minimum, this policy shall include:

a) stating whether the HEALTH SOFTWARE is compatible with the dependent component or operating system SECURITY update; and

b) for SECURITY updates that are unapproved by the HEALTH SOFTWARE MANUFACTURER, the mitigations that can be used in lieu of not applying the update.

Note: A dependent component is a component external to the HEALTH SOFTWARE on which the PRODUCT depends.

6.2.3 Timely delivery of SECURITY updates

The MANUFACTURER shall establish - as a part of the update activities - a policy that specifies the timeframes for delivering and qualifying (see 6.1.1) SECURITY updates to PRODUCT users. At a minimum, this policy shall consider the following factors:

a) the potential impact (technical, towards SAFETY, effectiveness, SECURITY) of the VULNERABILITY;

b) public knowledge of the VULNERABILITY;

c) whether published EXPLOITS exist for the VULNERABILITY;

d) the volume of deployed PRODUCTS that are affected; and

e) the availability of an effective mitigation when no HEALTH SOFTWARE update is being provided

Note 1: Some regulatory authorities may have specific timeframe requirements.

Note 2: The MANUFACTURER may categorize SECURITY updates (for example by potential impact) and specify appropriate timeframes.

Note 3: During an acceptable time-interval in which the MANUFACTURER develops a technical compensating control, any documented mitigations and constraints on the intended use can be based on RISK MANAGEMENT. It is advisable to develop and deploy a technical mitigation in HEALTH SOFTWARE.

Note 4: IEC TR 60601-4-5 specifies a minimum performance ("Essential Function" term as used in IEC 62443 series) to be available with medical devices in case of relevant cybersecurity attacks on the HDO IT network. Such minimum performance is needed to ensure basic functionality until a verified security update is available in situations in which all medical devices of the same type in the HDO may be affected by a given cybersecurity attack simultaneously. Therefore, for Medical Devices, the software lifecycle activities should ensure that

a) "essential functions" remain secure for the interval until a security update is installed, and

b) security updates should always re-establish the security capabilities as specified in the accompanying documents.

Additional guidance is provided by IEC TR 60601-4-5

6.3 Decommissioning and disposal of HEALTH SOFTWARE

The MANUFACTURER shall establish an ACTIVITY(S) to create PRODUCT user documentation that includes guidelines for removing the HEALTH SOFTWARE from use.

7 SECURITY RISK MANAGEMENT

7.1 Risk management Context

The MANUFACTURER shall establish and maintain a PROCESS for managing security risks related to Health Software as a part of product risk management. This PROCESS should consist of following PROCESS steps described in this chapter.

Note 1: SECURITY RISK MANAGEMENT can be conducted under the framework of ISO 14971 with an appropriate mapping of VULNERABILITY, THREAT and other SECURITY related terms. (See ISO/TR 24971:20XX for possible mapping.)

Note 2: See Annex D.

7.1.1 Product security context

The MANUFACTURER shall establish an ACTIVITY(S) to ensure that the intended product security context is documented.

This ACTIVITY(S) is required to ensure that the minimum requirements of the environment and the assumptions about that environment are documented in order to achieve the security level for which the product was designed. The purpose of defining this information is so that both the developers of the product and the product users have the same understanding about how the product is intended to be used. This will help the developers make appropriate design decisions and the users to use the product as it was intended.

Security context could include:

a) location in the network;
b) physical or cyber security provided by the environment where the product will be deployed;
c) isolation (from a network perspective); and
d) if known, potential impact to the environment (for example, loss of life, injury, loss of production, etc.)
e) security controls implemented in dedicated hardware with which the HEALTH SOFTWARE is intended to be used.

For example, it is important to document whether physical security is required. If no physical security is expected to be present, then that may add a number of related requirements such as not allowing pushbutton configuration on the product. Another example is if the product is expected to be protected by a user supplied firewall that connects it to the Health-IT-network or user network, the product would typically not require a firewall of its own.

Documenting these external security features for the product (its security context) allows developers to design a defense in depth strategy that complements this security context and testers to validate and verify the security of a product in an environment similar to how it should be deployed.

Having this process means that the deployment environment in which the product is intended to be used is correctly represented in all processes involved in the development and testing of this product and are documented

7.2 Identification of VULNERABILITIES, THREATS and associated adverse impacts

The MANUFACTURER shall establish an ACTIVITY(S) which identifies and documents any VULNERABILITIES, THREATS and associated adverse impacts affecting CONFIDENTIALITY, INTEGRITY, AVAILABILITY of ASSETS in HEALTH SOFTWARE –This Activity(s) shall consider the intended use and the INTENDED ENVIRONMENT OF USE with respect to the acceptable level of RESIDUAL RISK as defined in the Risk Management Process .

These activity(s) shall be employed to ensure that all products shall have a threat model specific to the current development scope of the product with the following characteristics (where applicable):

a) correct flow of categorized information throughout the system;
b) trust boundaries;
c) processes;
d) data stores;
e) interacting external entities;
f) internal and external communication protocols implemented in the product;
g) externally accessible physical ports including debug ports;
h) circuit board connections such as Joint Test Action Group (JTAG) connections or debug headers which might be used to attack the hardware;
i) potential attack vectors including attacks on the (intended) hardware, if applicable;
j) potential threats and their severity as defined by a vulnerability scoring system (for example, CVSS);
k) mitigations and/or dispositions for each threat;
l) security-related issues identified; and
m) external dependencies in the form of drivers or third-party applications (code that is not developed by the supplier) that are linked into the application.

The threat model shall be reviewed and verified by the development team to ensure that it is correct and understood.

The threat model shall be reviewed periodically (at least once a year) for released products and updated if required in response to the emergence of new threats to the product even if the design does not change.

Any issues identified in the threat model shall be addressed as defined in 9.4 and 9.5.

7.3 Estimation and evaluation of SECURITY Risk

The MANUFACTURER shall establish an ACTIVITY(S) to

a) estimate the risk of the items (Vulnerabilities, Threats) identified above. Risk estimation is done considering the SEVERITY of adverse impact of that item. This estimation can be supported by using VULNERABILITY scoring such as the Common Vulnerability Scoring System (CVSS)
b) evaluate the estimated risks and determine if the risk is acceptable or not.
c) inform the PRODUCT RISK MANAGEMENT PROCESS about any updates to the THREAT model.

7.4 Controlling SECURITY Risk

The MANUFACTURER shall determine risk control measures that are appropriate for reducing the risks to an acceptable level. The MANUFACTURER may consider the benefit-risk balance when determining whether the risk is acceptable

7.5 Monitoring the effectiveness of risk controls.

The manufacturer shall monitor the effectiveness of RISK CONTROLS by information collection and review during the post-market phase.

This ACTIVITY(S) shall also inform other ACTIVITIES and PROCESSES of the issue or related issue(s), including PROCESSES for other PRODUCTS / revisions; and inform third parties (e.g. suppliers) if problems have been found in third-party source code to be used with the HEALTH SOFTWARE.

Any issues identified in the THREAT model of released HEALTH SOFTWARE will be addressed as defined in 9.4 and 9.5.

8 Software configuration management PROCESS

The MANUFACTURER shall establish a general PRODUCT development/ MAINTENANCE /support PROCESS that includes configuration management with change controls and AUDIT LOGGING.

9 Software problem resolution PROCESS

9.1 Overview

The ACTIVITIES specified by this clause are used for handling SECURITY-related issues of HEALTH SOFTWARE.

9.2 Receiving notifications about VULNERABILITIES

The MANUFACTURER shall establish an ACTIVITY(S) that enables the reporting of information regarding VULNERABILITIES to the MANUFACTURER – independent of whether they come from an internal entity, an external entity or via a complaint-handling system.

This reception PROCESS shall receive and track to closure reports on SECURITY-related issues in the PRODUCT reported by internal (including an existing complaint-handling system) and external sources including at a minimum:

a) SECURITY VERIFICATION and VALIDATION testers;

b) suppliers of third-party components used in the PRODUCT;

c) PRODUCT developers and testers; and

d) PRODUCT users including integrators, operators, administrators, and maintenance personnel.

e) data obtained from AUDIT (EVENT) LOG information.

Note 1: Typically, such information comes from publications, reports, independent security research, internal investigation and so on.

Note 2: External SECURITY VERIFICATION and VALIDATION testers include researchers.

Note 3: Also see ISO/IEC 29147:2018 Information Technology – Security Techniques – Vulnerability Disclosure

9.3 Reviewing VULNERABILITIES

The MANUFACTURER shall establish an ACTIVITY(S) that enables the investigation of VULNERABILITIES in a timely manner to determine their:

a) applicability to the PRODUCT;

b) verifiability; and

c) related THREATS.

Note 1: Timeliness is driven by authorities, applicable legislation, regulatory policy and market forces.

Note 2: This PROCESS may be implemented for example as a part of the PROCESSES per ISO 13485:2016 clause 8.2.1 Feedback, 8.2.2 Complaint handling and 8.2.3 Reporting to regulatory authorities.

9.4 Analysing VULNERABILITIES

The MANUFACTURER shall establish an ACTIVITY(S) for analysing VULNERABILITIES in the PRODUCT to include:

a) assessing their impact with respect to:

1) the technical SECURITY context in which they were discovered; (see IEC 62443-4-1 Clause 6, Practice 2 – Specification of SECURITY requirements)
2) the PRODUCT'S INTENDED ENVIRONMENT OF USE; and
3) the PRODUCT'S defence in depth strategy;

b) SEVERITY as defined by a VULNERABILITY scoring system (for example, CVSS);
c) identifying all other PRODUCTS / PRODUCT versions containing the SECURITY-related issue (if any);
d) identifying the root cause of the issue;
e) identifying related SECURITY issues (that is, in the same PRODUCT) and
f) impact on PRODUCT SAFETY and effectiveness.

Note 1: For root cause analysis, a methodical approach such as method described in IEC 62740 may be employed.

Note 2: A root cause is the first event in a sequence of causal factors which is deviating from the intended sequence.

Note 3: Not all root causes can be fixed by technical measures in HEALTH SOFTWARE.

Note 4: This PROCESS may be implemented for example as a part of ISO 13485:2016 clause 8.5.2 and 8.5.3 .

9.5 Addressing SECURITY-related issues

The MANUFACTURER shall establish an ACTIVITY(S) to address SECURITY-related issues and determine whether to disclose them (under 10.4) based on the results of the impact assessment and the acceptable level of RESIDUAL RISK.

The MANUFACTURER shall establish an ACTIVITY(S) to determine whether and how identified SECURITY risks will be handled - via the Problem Resolution PROCESS or through updated specifications regarding the INTENDED ENVIRONMENT OF USE.

The MANUFACTURER shall establish an ACTIVITY(S) to review any changes to the design or implementation for impact on SAFETY, SECURITY and effectiveness.

The Manufacturer shall inform other processes of the issue or related issue(s), including processes for other products/product revisions.

The Manufacturer shall inform third parties if problems have been found in third-party source code to be used with the Health Software. In case of open-source software, the publishing platform may be used to inform about or fix the issue found.

This PROCESS shall include a periodic review of open SECURITY-related issues to ensure that issues are being addressed appropriately. This periodic review shall at a minimum occur during each PRODUCT release; see 10.3 "Continuous Improvement", and 10.5 "Periodic review".

Note 1: This periodic review may be implemented for example as a part of ISO 13485:2016 clause 8.2.6 Monitoring and measurement of PRODUCT

Note 1: As an example, an intended function including the transmission of PHI through an uncontrolled network may raise the need for data encryption.

Note 2:

- For some THREATS it can be feasible to not mitigate them through technical measures in HEALTH SOFTWARE, because they can be linked to the intended use or essential functions.

- Example: Console access to emergency / acute care devices would be hindered by overly complex authentication procedures and might delay the delivery of urgent care.

- Example: Hard cryptography algorithms for encrypting data used for near-field transmission in principle use considerable computing power and can drain the battery when implemented in smaller, mobile devices.

- Some THREATS can be better addressed by mitigations in the INTENDED ENVIRONMENT OF USE.

- There can be THREATS that are not exploitable because of measures in the design of the HEALTH SOFTWARE.

Note 3: Because of the complexity in determining the probability related to THREATS, the concept of likelihood is more appropriate and commonly used for IT- SECURITY. Likelihood of identified THREATS is typically expressed through structured scoring systems like Common Vulnerability Scoring Systems (CVSS), which may also take into account the attacker's gain in relation to the required effort.

Note 4: RISK MANAGEMENT for MEDICAL DEVICE SAFETY – as in ISO 14971 - can be supported by a THREAT MODELLING method to cover SECURITY THREATS.

Note 5: IEC 63069 explains the SAFETY / SECURITY relationship.

10 Quality management system

10.1 SECURITY expertise

The MANUFACTURER shall establish an ACTIVITY(S) for identifying and providing SECURITY training and assessment programs to ensure that personnel assigned to the organizational roles and duties specified in 4.1.2 have demonstrated SECURITY expertise appropriate for those PROCESSES. Results of this PROCESSES include role descriptions, training profiles and training records.

Note 1: This ACTIVITY(S) may be implemented for example as a part of ISO 13485:2016 clause 6.2 Human resources.

10.2 Components from third-party suppliers

The MANUFACTURER shall establish an ACTIVITY(S) to ensure that SECURITY LIFECYCLE PROCESSES that conform with this document are used by each third-party component supplier if that component can have an impact on SECURITY and

a) the supplier is contracted to develop components for a specific purpose for the MANUFACTURER, or

b) the component will be obtained as Off-the-shelf software

Note 1: This ACTIVITY(S) may be implemented for example as a part of ISO 13485:2016 clause 7.4 Purchasing.

10.3 Continuous improvement

The MANUFACTURER shall establish an ACTIVITY(S) for continuously improving the SDL (SECURITY development LIFECYCLE). This ACTIVITY(S) shall include the analysis of SECURITY defects in component/subsystem/system technologies that have been deployed to the field – due to insufficient or lacking ACTIVITIES.

Note 1: This PROCESS may be implemented for example as a part of ISO 13485:2016 clause 8.5 Improvement.

Note 2: This ACTIVITY(S) is required to ensure that the MANUFACTURER improve the rigor of their SECURITY activities over time. In case of process-dependent SECURITY defects it is important for the MANUFACTURER to help compensate for this by continuously improving their SECURITY activities.

10.4 Disclosing SECURITY-related issues

The MANUFACTURER shall establish an ACTIVITY(S) for informing regulatory authorities and PRODUCT users about reportable VULNERABILITIES (see 9.5) in supported PRODUCTS in a timely manner with content that includes but is not limited to the following information:

a) VULNERABILITY description, VULNERABILITY score as per CVSS or a similar system for ranking VULNERABILITIES, and affected PRODUCT version(s); and

b) description of the resolution.

Note 1: The description of the resolution can include references to installation of SECURITY updates - see IEC 62443-4-1, cl 12.

Note 2: Timeliness is driven by authorities, applicable legislation, regulatory policy, PRODUCT SAFETY and, market forces. The strategy for handling third-party component VULNERABILITIES discovered by the PRODUCT developer should take into account the possibility of premature public disclosure by the third-party component supplier.

Note 3: This PROCESS may be implemented for example as a part of ISO 13485:2016 clause 7.2.3.

Note 4: see Clauses 4.2, 6.2 and 10.6

10.5 Periodic review of SECURITY defect management practice

The MANUFACTURER shall establish an ACTIVITY(S) for conducting periodic reviews of the Software problem resolution PROCESS.

Periodic reviews of the ACTIVITIES shall, at a minimum, examine SECURITY-related issues managed through the PROCESS since the last periodic review to determine if the management PROCESS was complete, efficient, and led to the resolution of each SECURITY-related issue.

Periodic reviews of the SECURITY-related issue management PROCESS shall be conducted at least annually or as part of monitoring, measurement and analysis of PROCESSES of ISO 13485:2016 clause 4.1.3.

Note: This ACTIVITY(S) may be implemented for example as a part of ISO 13485:2016 clause 5.6, Management review.

10.6 ACCOMPANYING DOCUMENTS review

The MANUFACTURER shall establish an ACTIVITY(S) for identifying, characterizing and tracking to closure SECURITY-related errors and omissions in all user manuals including the SECURITY guidelines.

Note 1: This ACTIVITY(S) may be implemented for example as a part of ISO 13485:2016 clause 7.3, Design and development.

Annex A (informative)
Rationale

IEC 62443 is a series of Industrial Automation and Controls Systems SECURITY specifications. This series is the successor of ISA-99 and a well-recognized set of SECURITY standards.

Industrial Automation and Control Systems (IACS) have a recognition for SAFETY and effectiveness, which are key properties that are also applicable to the field of HEALTH SOFTWARE.

This document addresses LIFECYCLE PROCESSES in responsibility of the MANUFACTURER and therefore takes the requirements of IEC 62443-4-1, as far as they…

- are relevant for HEALTH SOFTWARE
- specify a PROCESS-related requirement
- address the MANUFACTURER
- do not specify product capabilities
- do not specify documentation content for accompanying documents – which will be specified by IEC TR 60601-4-5

In order to extend existing LIFECYCLE PROCESSES for HEALTH SOFTWARE, these requirements have been arranged in a structure reflecting that of IEC 62304.

Implementation of IEC 62304 is not required for implementing the processes specified in this document. However, if a manufacturer identifies in their processes those activities specified in IEC 62304, it is easier to determine the related ACTIVITY(S) towards information security specified in this document.

The secure coding best practices for HEALTH SOFTWARE should include at a minimum:

a) avoidance of potentially exploitable implementation constructs – implementation design patterns that are known to have SECURITY WEAKNESSES;

b) avoidance of banned functions and coding constructs/design patterns – software functions and design patterns that should not be used because they have known SECURITY WEAKNESSES;

   Note: For common libraries and programming languages there are public lists of banned functions. Per secure coding standards, the MANUFACTURER can decide these or more functions.

c) automated tool use and settings (for example, for static analysis tools);

d) general secure coding practices (for example external secure coding standards);

e) validity checking of all inputs that cross a trust boundary.

f) error handling.

The MANUFACTURER should evaluate each type of alert from static analysis whether it justifies a code change.

The application of secure coding standards can be judged on a case-by-case basis.

Annex B
(informative)
Guidance on Implementation of SECURITY Lifecycle Activities

B.1 Overview

Information security of a PRODUCT containing software can be supported by functional SECURITY CAPABILITIES of that software – typically implementing protection from, detection of, response to and recovery from incidents that may compromise the CONFIDENTIALITY, INTEGRITY or AVAILABILITY of the PRODUCT'S ASSETS.

Although this standard focuses on software there are additional SECURITY considerations for the physical device that the software is running on that should be included in all process activities. Examples are to reduce physical interface ports, like JTAG or unused USB ports, similar to limiting open network ports on a software level. Similarly, there are mitigations provided by the device, such as physical locks to provide access control to internal media.

The technical reports IEC TR 60601-4-5 and IEC TR 80001-2-2 give guidance towards the identification and communication of such SECURITY CAPABILITIES. While these technical reports address medical devices, their concepts and measures can easily be transferred to HEALTH SOFTWARE.

Another aspect is related to the lifecycle: MANUFACTURERS of HEALTH SOFTWARE can establish PROCESSES that avoid or mitigate VULNERABILITIES or reduce their impact to the PRODUCTS' INTENDED PURPOSE. Some PROCESSES – for instance requirements engineering and, THREAT RISK ANALYSIS - link the perspective of functions with the view on processes. It is important to understand that only the combination of both PRODUCT capabilities as well as measures in the LIFECYCLE PROCESSES can provide effective information security.

B.2 THREAT / RISK ANALYSIS (TRA)

Security incidents may affect the PRODUCT'S SAFETY or effectiveness. The specific relationship between VULNERABILITIES and risks regarding SAFETY or effectiveness depends on the design, implementation and purpose of the respective PRODUCT. A PRODUCT risk analysis for SAFETY therefore must consider the effects of VULNERABILITIES to the key functions of the product. As a part of that ACTIVITY(S), THREAT / RISK ANALYSIS is performed for HEALTH SOFTWARE.

Those scenarios with an attacker exploiting a known vulnerability may be considered as "foreseeable" with respect to product risk management. In case the USE ENVIRONMENT or other mitigation controls might fail to prevent a certain type of attack, that scenario becomes "foreseeable" from the MANUFACTURER'S perspective. TRA identifies and evaluates such threat scenarios – taking into account

a) the dedicated hardware with which the HEALTH SOFTWARE is intended to be used.

b) the intended operational context and

c) the potential data/control flows from external actors into the HEALTH SOFTWARE.

B.3 Threat / Risk Management

One outcome of applying TRA is an evaluation of known vulnerabilities that may affect the HEALTH SOFTWARE'S ASSETS (data, software functions, software services) with respect to CONFIDENTIALITY, INTEGRITY or AVAILABILITY – and how that is related to the overall safety, security and effectiveness of the product as a whole.

The scenarios considered during TRA is not limited to those foreseen by the PRODUCT'S intended use, however TRA takes the intended operational environment into account.

Options for determining the acceptable RESIDUAL RISK with remaining VULNERABILITIES include one or more of the following:

a) fixing the issue through one or more of the following:

1) defence in depth strategy or design change;
2) addition of one or more security requirements and/or capabilities;
3) use of compensating mechanisms; and/or
4) disabling or removing features; with respect to the safe and effective use of HEALTH SOFTWARE;

b) creating a remediation plan to fix the problem;

c) deferring the problem for future resolution (reapply this requirement at some time in the future) and specifying the reason(s) and associated risk(s);

d) not fixing the problem if the residual risk is below the established acceptable level of residual risk.

When the resolution decision is to fix the security-related issue in the PRODUCT implementation, the timing of the release of the fix can result in a security update to be deferred until the next release.

B.4 Software Development Planning

B.4.1 Development process

An appropriate development PROCESS for HEALTH SOFTWARE should implement a development/ maintenance/ support process as required in IEC 62304 and additionally implement items of the list specified in 5.1.1.

B.4.2 Development environment security

HEALTH SOFTWARE must be protected from any compromises via the development environment. For instance, the introduction of malicious software or the theft of credentials such as software signing certificates.

B.5 Health Software Requirements Analysis

B.5.1 HEALTH SOFTWARE security requirements

This requirement assumes that a system at a higher level has already defined a security level for this (sub)system. This is described per IEC 62443-3-2 in general and via IEC TR 60601-4-5 for PEMS.

B.5.2 Security requirements review

The implementation of SECURITY CAPABILITIES may have an impact on the PRODUCT'S SAFETY or effectiveness. This review may determine an appropriate requirement towards implementing SECURITY CAPABILITIES in a balanced way.

B.6 Software Architectural Design

B.6.1 Defense in depth architecture/design

Defense in depth is an approach to information security in which a series of defensive mechanisms are layered in order to protect information ASSETS: If one mechanism fails, another layer will thwart an attack. This multi-layered approach with intentional redundancies increases the security of a system as a whole and addresses many different attack vectors. Defense in depth is commonly referred to as the "castle approach" because it mirrors the layered defenses of a medieval castle.

Defense-in-depth reduces the likelihood of attacks to succeed, it reduces the impact of attacks and allows the target system to take compensating actions.

B.6.2 Secure design principles

The principles described in this requirement are relevant to the design of any system, whether for apps, client or server, cloud-based services, or Internet-of-Things devices. The specifics of

their application will vary – a cloud service may require multiple administrative roles, each with its own least privilege, while an IoT device will require special considerations of the need for security updates and of the need to fail securely and safely.

However, the principles are general and provide valuable security guidance for the designers and architects of all classes of systems. Elements of such a process need additional specifications that depend on the programming environment and the information technology used. There are specifications from Standard-Developing Organisations (SDOs) or associations with more detailed specifications (potentially depending on technology or context).

B.6.3 Security architectural design review

The ability of an architecture to ensure stable and predictable behavior is important, because adverse conditions can come intentionally or unintentionally; they may show up via adverse calls / data when HEALTH SOFTWARE is being used in its USE ENVIRONMENT.

B.7 Software Unit Implementation and Verification

B.7.1 Secure Coding Standards

Secure coding standards should incorporate the following principles:

- Establish coding standards and conventions
- Use safe functions only
- Use current compiler and toolchain versions and secure compiler options
- Handle input and other data safely (i.e. in a restrictive, cautious way…)
- Use source code analysis tools to find security issues early
- Handle errors

From 5.5.2

Note 1: Source Code Analysis (SCA) detects the potential for errors such as buffer overflows, null pointer dereferencing, and similar.

Note 2: SCA can be done using a tool if one is available for the language used. In addition, static code analysis can be done on all source code changes including new source code.

B.7.2 Secure implementation

The manufacturer may foresee an architecture and design that allow for updating or substituting cryptographic technology – for example. The goal here is to implement with cryptographic agility in mind: encryption algorithms might potentially be broken at any time, even if they are considered current best practices, and encryption libraries may have vulnerabilities that undermine otherwise sound algorithms. In the example, a secure implementation should ensure, that some encryption strategy specifies how applications and services should implement their encryption to enable transition to new cryptographic mechanisms, libraries and keys when the need arises.

B.7.3 Security testing

The manufacturer may foresee a (semi-)independent internal testing team and/or the use of a third party security test organization. Individuals who are independent from the developers who designed and implemented the security features must do the security testing. Section 5.7 on Software System Testing provides the requirements and more detail related to security testing. An overview of some automated and manual testing techniques includes:

B.7.3.1 Vulnerability scanning

Vulnerability scanning is the automated detection of known vulnerabilities. Scanners will detect installed software, open network ports, operating system configuration and other security relevant information. Many vulnerability scanners allow for both authenticated and unauthenticated scans. An authenticated scan means that the tool has administrative system

credentials to bypass certain protections and will be able to assess the systems configuration with much more detail and accuracy. OWASP maintains a list of Vulnerability Scanning Tools.

B.7.3.2 Input validation testing

Input validation testing tries to detect undesired system behavior when incorrect data or excessive load of data are send to a system interface. Often automated tools are used and the more specialized the tool is for a certain interface protocol, the more accurate the test results will be. Examples are fuzz testing, buffer overflow and format error testing. Specialized injection testing techniques exists for protocols such as SQL, LDAP, XML and cross-site scripting.

B.7.3.3 Penetration testing

Penetration testing, also called pen-testing, focuses specifically on compromising confidentiality, integrity or availability. It can involve defeating multiple aspects of the defense in depth design. For example, bypassing authentication to access the product, using elevation of privilege to gain administrative access and then compromising confidentiality by breaking encryption. As this example shows, penetration testing involves approaching testing like an attacker and often involves exploiting chained vulnerabilities in a product using both tools and manual skills. Results of the vulnerability scanning and other tests could provide valuable input to develop manual attack scenarios.

B.8 Software Release

B.8.1 Security measures expected in the environment

In HEALTH SOFTWARE there may be VULNERABILITIES for which compensating controls could be inappropriate considering the safety and effectiveness of the product when used as intended.

The product should anticipate its intended environment of use to a certain extent. A declaration of external controls expected to be provided can be used to define shared responsibilities – for example as specified in IEC TR 60601-4-5 and in IEC TR 80001-2-2 for which a well-established guidance is published as ANSI MDS2.

Annex C
(informative)
References to other standards

ISO 27000 Information technology -- Security techniques -- Information security management systems -- Requirements

ISO 27034-3 Information technology -- Application security -- Part 3: Application security management process

ISO 27799 Health informatics -- Information security management in health using ISO/IEC 27002

IEC TR 60601-4-5 Medical electrical equipment – Part 4-5 Guidance and interpretation – Safety related technical security specifications for medical devices

IEC 62304 Ed. 2: Health software - Software Lifecycle processes

IEC 62443-3-2 Security Risk Assessment and System Design

IEC 62443-3-3 System Requirements and Security Levels

IEC 62443-4-1 Secure Product Development Lifecycle Requirements

IEC 62443-4-2 Technical Security Requirements for IACS Components

IEC TR 80001-2-2 Application of Risk Management for IT-Networks incorporating Medical Devices – Part 2-2: Guidance for the Disclosure and Communication of Medical Device Security Needs, Risks and Controls,

IEC/ISO 80001-2-8 Application of Risk Management for IT-Networks incorporating Medical Devices – Part 2-8: Application guidance - Guidance on Standards for Establishing the Security Capabilities identified in IEC TR 80001-2-2

ISO/IEC 81001-1 Health software and health IT systems safety, effectiveness and security - Foundational principles, concepts and terms

IEC 82304-1 Health software - Part 1: General requirements for product safety.

Note: IEC 82304-1 specifies safety requirements for Health Software products (“stand-alone”).

Annex D
(informative)
THREAT MODELLING

D.1 General

THREAT MODELLING is a systematic approach for analyzing the SECURITY of an item in a structural way such that VULNERABILITIES can be identified, enumerated, and prioritized, all from a hypothetical attacker's point of view. THREAT MODELING can be applied to a wide range of things, including software, devices, systems, networks, distributed systems and business PROCESSES. THREAT MODELING typically employs a systematic approach to identify ATTACK vectors and ASSETS most desired by an attacker. This leads to a decomposition of the item (software, device, system, and so on) to look at each possible ATTACK vector and ASSET individually and determine to which kind of ATTACKS they are vulnerable. From this, a list of VULNERABILITIES can be created and ordered in terms of risk, potential to impact SAFETY, effectiveness, or any other criteria deemed appropriate (like privacy).

There are various approaches to creating a THREAT model that range from making a list of known VULNERABILITIES to adopting a framework, some examples:

D.2 ATTACK-Defense Trees

An ATTACK-Defense Tree (ADTree) is a node-labeled rooted tree describing the measures an attacker might take to ATTACK a system and the defenses that a defender can employ to protect the system.

D.3 CAPEC / OWASP / SANS

A basic approach is to use lists of known top THREATS such as the OWASP Top 10 or the CWE/SANS Top 25. The Common Attack Pattern Enumeration and Classification (CAPEC) has a more comprehensive dictionary of known patterns of ATTACK employed by adversaries to exploit known WEAKNESSES.

D.4 CWSS

The Common Weakness Scoring System (CWSS) both identifies VULNERABILITIES and provides a scoring system to prioritize them. It is a collaborative, community-based effort that focuses on analyzing software and reported bugs to determine the relative importance of the detected WEAKNESSES.

D.5 DREAD

DREAD is a classification scheme for quantifying, comparing and prioritizing the amount of risk presented by each evaluated THREAT. DREAD modelling focuses on risk rating. The DREAD algorithm is used to compute a risk value, which is an average of all five categories: **D**amage, **R**eproducibility, **E**xploitability, **A**ffected Users, and **D**iscoverability.

D.6 List Known Potential VULNERABILITIES

One may attempt listing all the VULNERABILITIES that could affect your system. While it is impossible to list all potential VULNERABILITIES, one should concentrate those VULNERABILITIES that could be exercised by known THREATS.

D.7 OCTAVE

OCTAVE is a heavyweight risk methodology approach originating from Carnegie Mellon University's Software Engineering Institute (SEI) in collaboration with CERT. OCTAVE focuses on organizational risk, not technical risk.

D.8 STRIDE

STRIDE is a model for system decomposition, by characterizing known THREATS according to the kinds of EXPLOITS used. The STRIDE acronym stands for each of the categories: **S**poofing,

Tampering, **R**epudiation, **I**nformation Disclosure, **D**enial of Service, and **E**levation of Privilege. STRIDE does not include a scoring system.

D.9 Trike

Trike is a THREAT MODELING framework with similarities to the STRIDE and DREAD THREAT MODELLING PROCESSES. Trike differs in that it uses a risk-based approach with distinct implementation, THREAT, and risk models, instead of using the STRIDE/DREAD aggregated THREAT model (ATTACKs, THREATS, and WEAKNESSES).

D.10 VAST

VAST is an acronym for Visual, Agile, and Simple THREAT MODELING. The principle of this approach is the necessity of scaling the THREAT modeling PROCESS across the infrastructure and entire software development LIFECYCLE. The approach integrates into an Agile software development methodology. The methodology provides an application and infrastructure visualization scheme such that the creation and use of THREAT models do not require specific SECURITY subject matter expertise.

Annex E
(informative)
Relation to practices in IEC 62443-4-1

E.1 ISO/IEC 80001-5-1 to IEC 62443-4-1:2018

4.1.1	Not in 62443-4-1
4.1.2	SM-2
4.1.3	SM-3
4.2	Not in 62443-4-1
5.1.1	SM-1, SM-5
5.1.2	SM-7
5.1.3	SI-2
5.2.1	SR-3
5.2.1	SR-4
5.2.2	SR-5
5.2.2	SVV-5
5.2.3	SM-9
5.3.1	SD-2
5.3.2	SD-4
5.3.3	Not in 62443-4-1
5.4.1	Not in 62443-4-1
5.4.2	SD-1
5.4.3	SD-3
5.5.1	Not in 62443-4-1
5.5.2	SI-1
5.6	Not in 62443-4-1
5.7.1	SVV-1
5.7.2	SVV-2
5.7.3	SVV-3
5.7.4	SVV-4
5.8.1	Not in 62443-4-1
5.8.2	SG-5
5.8.2	SG-6
5.8.3	SM-6
5.8.4	SM-8
5.8.5	SM-11
5.8.6	SM-12
6.1.1	SUM-1
6.1.2	SUM-4
6.2.1	SUM-2
6.2.2	SUM-3
6.2.3	SUM-5
6.3	SG-4
7.1.1	SR-1
7.2	SR-2
7.3	Not in 62443-4-1
7.4	Not in 62443-4-1
7.5	Not in 62443-4-1
8	SM-1
9.2	DM-1
9.3	DM-2
9.4	DM-3
9.5	SM-4
10.1	SM-4
10.2	SM-10
10.3	SM-13
10.4	DM-5
10.5	DM-6
10.6	SG-7
Annex A	SI-2

Note: A specification of this document mapped to “not in 62443-4-1” should still reflect the intentions of IEC 62443-4-1.

E.2 IEC 62443-4-1:2018 to ISO/IEC 80001-5-1

Note that SG-# requirements are not included as stated in the purpose section 1.1 and in Annex A (rationale), this document excludes specification of ACCOMPANYING DOCUMENT contents.

SM-1	5.1.1, 8
SM-2	4.1.2
SM-3	4.1.3
SM-4	10.1
SM-5	5.1.1
SM-6	5.8.3
SM-7	5.1.2
SM-8	5.8.4
SM-9	5.2.3
SM-10	10.2
SM-11	5.8.5
SM-12	5.8.6
SM-13	10.3
SR-1	7.1.1
SR-2	7.2
SR-3	5.2.1
SR-4	5.2.1
SR-5	5.2.2
SD-1	5.4.2
SD-2	5.3.1
SD-3	5.4.3
SD-4	5.3.2
SI-1	5.5.2
SI-2	5.1.3, Annex A

SVV-1	5.7.1
SVV-2	5.7.2
SVV-3	5.7.3
SVV-4	5.7.4
SVV-5	5.2.2
DM-1	9.2
DM-2	9.3
DM-3	9.4
SM-4	9.5
DM-5	10.4
DM-6	10.5
SUM-1	6.1.1
SUM-2	6.2.1
SUM-3	6.2.2
SUM-4	6.1.2
SUM-5	6.2.3
SG-1	-
SG-2	-
SG-3	-
SG-4	6.3
SG-5	5.8.2
SG-6	5.8.2
SG-7	10.6

Annex F
(informative)
Document specified in IEC 62443-4-1

This annex specifies product-related documents which support the secure use of HEALTH SOFTWARE.
A manufacturer claiming conformance towards IEC 62443-4-1 can demonstrate conformance to this document including this Annex on product-related documentation.
The processes specified by this Annex are used to provide documentation that describes how to integrate, configure and maintain the defense-in-depth strategy of the HEALTH SOFTWARE IN accordance with its security context. Applying and maintaining the defense-in-depth strategy for a specific HEALTH SOFTWARE installation will typically address the following:

1) Policies and procedures associated with the HEALTH SOFTWARE security context,
2) Architectural considerations, such as firewall placement and the use of compensating mechanisms including security measures
3) Configuring security settings/options such as configuring firewall rules and managing user accounts,
4) Use of tools to assist in hardening HEALTH SOFTWARE

F.1 Release Documentation

F.1.1 HEALTH SOFTWARE defense in depth documentation

The MANUFACTURER shall establish an ACTIVITY to create HEALTH SOFTWARE documentation that describes the compensating controls for the HEALTH SOFTWARE to support installation, operation and MAINTENANCE that includes:

a) SECURITY capabilities implemented by the HEALTH SOFTWARE and their role in the defense in depth strategy;
b) THREATS addressed by the defense in depth strategy;
c) HEALTH SOFTWARE user mitigation strategies for known SECURITY risks associated with the HEALTH SOFTWARE, including risks associated with LEGACY SOFTWARE.

Note 1: IEC TR 60601-4-5 gives guidance on the specification of SECURITY capabilities and their documentation in the accompanying documents and provides a method of determining requirements from the SECURITY CAPABILITY level.

Note 2: IEC TR 80001-2-2 specifies SECURITY-related needs, risks and controls as a guidance for disclosure and communication between the MANUFACTURER and the HEALTHCARE DELIVERY ORGANIZATION.

F.1.2 Defense in depth measures expected in the environment

The MANUFACTURER shall establish an ACTIVITY to create PRODUCT documentation that declares external SECURITY controls expected to be provided or implemented by the external environment.

Note: Software should not be placed on the market based on the assumption that all users have some certain technical control in place.

F.1.3 SECURITY hardening guidelines

The MANUFACTURER shall establish an ACTIVITY to create HEALTH SOFTWARE documentation that includes guidelines for hardening the HEALTH SOFTWARE when deploying, installing and maintaining the HEALTH SOFTWARE. If applicable, the guidelines shall include but are not limited to, instructions, rationale and recommendations for the following:

a) Integration of the HEALTH SOFTWARE, including third-party components, into its HEALTH SOFTWARE SECURITY context
b) Integration of the HEALTH SOFTWARE'S application programming interfaces/protocols with user applications;

c) Applying and maintaining the HEALTH SOFTWARE'S defense in depth strategy
d) Configuration and use of SECURITY options/capabilities in support of local SECURITY policies, and for each SECURITY option/ CAPABILITY:
    1) Its contribution to the HEALTH SOFTWARE'S defense in depth strategy
    2) Descriptions of configurable and default values that includes how each affects SECURITY along with any potential impact each has on work practices; and
    3) Setting/changing/deleting its value;
e) Instructions and recommendations for the use of all SECURITY-related tools and utilities that support administration, monitoring, incident handling and evaluation of the SECURITY of the HEALTH SOFTWARE;
f) Instructions and recommendations for periodic SECURITY MAINTENANCE activities;
g) Instructions for reporting SECURITY incidents involving the HEALTH SOFTWARE to the MANUFACTURER; and
h) Description of the SECURITY best practices for MAINTENANCE and administration of the HEALTH SOFTWARE.

F.2 Documents for Decommissioning Health Software

The guidelines for Health Software Decommissioning shall include, but are not limited to instructions and recommendations for the following:

a) removing the HEALTH SOFTWARE from its INTENDED ENVIRONMENT OF USE (see IEC 62443-4-1, Clause 6, Practice 2 – Specification of SECURITY requirements);
b) removing patient and configuration data stored within the environment;
c) secure transfer, migration, archiving and deletion of data stored in the HEALTH SOFTWARE; and
d) secure disposal of the HEALTH SOFTWARE to prevent potential disclosure of data contained in the HEALTH SOFTWARE that could not be removed as described in c) above.