



DPC: 25 / 30502360 DC

BSI Group Headquarters

389 Chiswick High Road London W4 4AL

Tel: +44 (0)20 8996 9000

Fax: +44 (0)20 8996 7400

www.bsigroup.com

Date: 30 January 2025

Origin: International

Latest date for receipt of comments: 12 February 2025

Project No. 2024/03021

Responsible committee: CH/62 Medical equipment, software, and systems

Interested committees:

Title: BS EN IEC 62304 Health software – Software life cycle processes

Please notify the secretary if you are aware of any keywords that might assist in classifying or identifying the standard or if the content of this standard

- i) has any issues related to 3rd party IPR, patent or copyright
- ii) affects other national standard(s)
- iii) requires additional national guidance or information

**WARNING: THIS IS A DRAFT AND MUST NOT BE REGARDED OR USED AS A BRITISH STANDARD.
THIS DRAFT IS NOT CURRENT BEYOND 12 February 2025**

This draft is issued to allow comments from interested parties; all comments will be given consideration prior to publication. No acknowledgement will normally be sent. **See overleaf for information on the submission of comments.**

No copying is allowed, in any form, without prior written permission from BSI except as permitted under the Copyright, Designs and Patent Act 1988 or for circulation within a nominating organization for briefing purposes. Electronic circulation is limited to dissemination by e-mail within such an organization by committee members.

Further copies of this draft may be purchased from BSI Shop <http://shop.bsigroup.com> or from BSI Customer Services, Tel: +44(0) 20 8996 9001 or email cservices@bsigroup.com. British, International and foreign standards are also available from BSI Customer Services.

Information on the co-operating organizations represented on the committees referenced above may be obtained from <http://standardsdevelopment.bsigroup.com>

Responsible Committee Secretary: Ms Lindsey Ferrari (BSI)

Direct tel:

E-mail: lindsey.ferrari@bsigroup.com

Introduction

This draft standard is based on national and international discussions. Your comments on this draft are invited and will assist in the preparation of the consequent standard.

For international standards, comments will be reviewed by the relevant UK national committee before sending the consensus UK vote and comments to the international committee, which will then decide appropriate action. If the international standard is approved, it is usual for the text to be published as a British Standard.

For national standards, comments will be reviewed by the relevant UK national committee and the resulting standards published as a British Standard.

UK Vote

Please indicate whether you consider the UK should submit a negative (with supporting technical reasons) or positive vote on this draft. Please indicate if you are aware of any reason why this draft standard should not be published as a British Standard.

Submission of Comments

- **Annotated drafts are not acceptable and will be rejected.**
- All comments should be submitted online at <https://standardsdevelopment.bsigroup.com/>. You will need to register in order to comment.

Template for comments and secretariat observations

Date: xx/xx/20xx	Document: ISO/DIS xxxx
------------------	------------------------

1	2	(3)	4	5	(6)	7
MB	Clause No./ Subclause No./Annex (e.g. 3.1)	Paragraph/Figure/ Table/Note	Type of comment	Commend (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
	3.1	Definition 1	ed	Definition is ambiguous and needs clarifying.	Amend to read '...so that the mains connector to which no connection...'	
	6.4	Paragraph 2	te	The use of the UV photometer as an alternative cannot be supported as serious problems have been encountered in its use in the UK.	Delete reference to UV photometer.	



PROJECT NUMBER:

IEC 62304 ED2

DATE OF CIRCULATION:

2025-01-17

CLOSING DATE FOR COMMENTS:

2025-03-14

SUPERSEDES DOCUMENTS:

62A/1553/RR

IEC SC 62A : COMMON ASPECTS OF MEDICAL EQUIPMENT, SOFTWARE, AND SYSTEMS

SECRETARIAT:

United States of America

SECRETARY:

Ms Colleen Elliott

OF INTEREST TO THE FOLLOWING COMMITTEES:

TC 62, SC 62B, SC 62C, SC 62D, TC 66

HORIZONTAL FUNCTION(S):

ASPECTS CONCERNED:

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.

Recipients of this document are invited to submit, with their comments, notification of any local regulations or technical reasons that may exist and should be considered should this proposal proceed, recognizing that failure to address such requirements could result in the need for new or existing "In Some Countries" clauses. (SEE [AC/22/2007](#) OR [NEW GUIDANCE DOC](#)).

TITLE:

Health software – Software life cycle processes

NOTE FROM TC/SC OFFICERS:

IEC/SC 62A/MT 49 has prepared rationales for design decisions and clause modifications (additions, deletions, revisions) made during the development of this draft of IEC 62304 Ed. 2.0. Please refer to this document (SC62A/MT49/N0166) while preparing comments. Also, please refer to the design specification for the Second Edition of IEC 62304 at:

<https://assets.iec.ch/public/sc62a/IEC%2062304%20Ed.%202.0%20Design%20Specification.pdf?2025010923>.

NOTE: It is critical to fully read and follow the complete instructions provided in SC62A/MT49/N0166 to properly comment on the CD fragments for the comments to be accepted.

Copyright © 2025 International Electrotechnical Commission, IEC. All rights reserved. It is permitted to download this electronic file, to make a copy and to print out the content for the sole purpose of preparing National Committee positions. You may not copy or "mirror" the file or printed version of the document, or any part of it, for any other purpose without permission in writing from IEC.

Link to Committee Draft (CD) online document:

<https://osd.iec.ch/#/editor/archive/15951239-4bf5-ce35-e063-1510000a5d03/en/CD/1>

How to access

This link will lead you to the Online Standards Development (OSD) platform for national mirror committee's comments. It replaces the project content, which is usually circulated as a PDF.

For OSD NC Managers

If not already done, your respective National Committee Administrator will create your profile via Symphony (<https://symphony.iec.ch>) as an OSD NC Manager to oversee the comment resolution process for the preparation of your national comments. Once you have been granted access via Symphony, please use your usual IEC credentials to access the link.

If you do not have access, please contact your National Committee Administrator or our OSD support team.

For OSD NC Commenters

If not already done, your OSD NC Manager will create your profile via Symphony as an OSD NC Commenter. Once you have been granted access via Symphony, please use your usual IEC credentials to access the link. You may then provide your comments directly in the OSD.

As an alternative solution, OSD NC Commenters who wish to continue using the IEC comment form can use the regular template together with a PDF of the draft document downloadable from the OSD platform. However, please note that line numbers are no longer available, and it is not possible to add tables and graphics or use track changes.

For Voters

Once all your national comments have been resolved, it is no longer necessary to upload the comment form to the IEC Voting System anymore. Instead, please use the following radio buttons in the IEC Voting System to indicate whether or not your National Committee has submitted comments in the OSD:

Comment: ☐ No Comment at this stage ☐ Comments in

Reference materials

We recommend that OSD NC Managers and OSD NC Commenters review the available materials to better understand the member commenting on the OSD platform. This includes:

- latest videos and presentations on the [OSD webpage, and](#)
- step-by-step explanations on [Helpscout](#).

Contact

Should you require any assistance, please contact the OSD project team at osd_support@iec.ch.

CONTENTS

FOREWORD	7
INTRODUCTION	9
1 Scope	15
1.1 Purpose	15
1.2 Field of application	15
1.3 Relationship to other standards	16
1.4 Conformance	16
2 Normative references	16
3 Terms and definitions	16
4 General requirements	22
4.1 Not used	22
4.2 risk management	22
4.3 Software process rigor level	22
4.3.1 software system	22
4.3.2 Decomposition into software item	24
5 Software development process	24
5.1 Software development planning	24
5.1.1 Software development plan	24
5.1.2 Keep software development plan updated	25
5.1.3 Software development plan reference to system design and development	25
5.1.4 Software development standards, methods and tools planning	25
5.1.5 Software integration and integration testing planning	25
5.1.6 Software verification planning	25
5.1.7 Software risk management planning	26
5.1.8 Documentation planning	26
5.1.9 Software configuration management planning	26
5.1.10 Supporting items to be controlled	26
5.1.11 Software configuration item control before verification	27
5.1.12 Not used	27
5.1.13 Software release planning	27
5.1.14 Software maintenance planning	27
5.1.15 artificial intelligence (capability) (AI) planning	27
5.1.16 Communication plan	30
5.2 Software requirements analysis	30
5.2.1 Define and document software requirements from system requirements	30
5.2.2 Software requirements content	30
5.2.3 Include risk control measures in software requirements	31
5.2.4 Re-evaluation product risk analysis	31
5.2.5 Update requirements	31
5.2.6 Verify software requirements	32
5.3 Software architecture design	32
5.3.1 Transform software requirements into an architecture	32

5.3.2	Develop an architecture for the interfaces of software item	32
5.3.3	Specify functional and performance requirements of soup item	32
5.3.4	Specify system hardware and software required by soup item	32
5.3.5	Identify segregation necessary for risk control	32
5.3.6	verification software architecture	33
5.4	Software design.....	33
5.4.1	Subdivide software into software unit	33
5.4.2	Develop design for each software unit	33
5.4.3	Develop design for interfaces	33
5.4.4	verification design	33
5.5	software unit implementation	33
5.5.1	Implement each software unit.....	33
5.5.2	Establish software unit verification process	34
5.5.3	software unit acceptance criteria	34
5.5.4	Additional software unit acceptance criteria.....	34
5.5.5	software unit verification.....	34
5.5.6	Static Code Analysis	34
5.6	Software integration and integration testing	34
5.6.1	Integrate software unit.....	34
5.6.2	Verify software integration.....	35
5.6.3	Software integration testing	35
5.6.4	Software integration testing content	35
5.6.5	evaluation software integration test procedures	35
5.6.6	Conduct regression tests.....	35
5.6.7	Integration test record contents	35
5.6.8	Use software problem resolution process	36
5.7	software system testing	36
5.7.1	Establish tests for software requirements	36
5.7.2	Use software problem resolution process	36
5.7.3	Retest after changes	36
5.7.4	evaluation software system testing	36
5.7.5	software system test record contents	36
5.8	Software release	37
5.8.1	Ensure software verification is complete.....	37
5.8.2	Document known residual anomaly	37
5.8.3	evaluation known residual anomaly	37
5.8.4	Document released version	37
5.8.5	Document how released software was created	37
5.8.6	Ensure activities and tasks are complete	37
5.8.7	Archive software.....	38
5.8.8	Ensure reliable delivery of released software	38
6	Software maintenance process.....	38
6.1	Establish software maintenance plan	38
6.2	Problem and modification analysis—	39
6.2.1	Document and evaluation feedback	39
6.2.2	Use software problem resolution process	40
6.2.3	Analyse change request	40
6.2.4	change request approval	40

6.2.5	Communicate to users and regulators	40
6.3	Modification implementation	40
6.3.1	Use established process to implement modification	40
6.3.2	Re-release modified software system	41
6.4	health software disposal	41
6.4.1	Disposal strategy	41
6.4.2	Performing disposal	41
7	Software risk management process	41
7.1	Analysis and control of software contributing to hazardous situations	41
7.1.1	Identify software items that could contribute to a hazardous situation	41
7.1.2	Identify potential causes of contribution to a hazardous situation	42
7.1.3	Evaluate published soup anomaly lists	42
7.1.4	Document potential causes	42
7.1.5	Define risk control measures	42
7.1.6	risk control measures implemented in software	43
7.2	verification of risk control measures	43
7.2.1	Verify risk control measures	43
7.2.2	Document traceability	43
7.3	risk management of software changes	43
7.3.1	Analyse changes to medical device software with respect to safety	43
7.3.2	Analyse impact of software changes on existing risk control measures	43
7.3.3	Perform risk management activity based on analyses	43
8	Software configuration management process	43
8.1	Configuration identification	44
8.1.1	Establish means to identify configuration item	44
8.1.2	Identify soup	44
8.1.3	Identify system configuration documentation	44
8.2	Change control	44
8.2.1	Change configuration items	44
8.2.2	Implement changes	44
8.2.3	Verify changes	44
8.2.4	Provide means for traceability of change	45
8.3	Configuration status accounting	45
9	Software problem resolution process	45
9.1	Prepare software problem report	45
9.2	Investigate the problem related to health software	45
9.3	Not used	46
9.4	Use change control process	46
9.5	Maintain records	46
9.6	Analyse problems for trends	46
9.7	Verify software problem resolution	46
9.8	Test documentation contents	46
Annex A (informative)	Rationale for the requirements of this standard	48
A.1	Rationale	48
A.2	Summary of requirements by software process rigor level	49
A.3	Adequacy of testing	50

A.3.1	Motivation	50
A.3.2	Measurement of Test Adequacy	50
A.3.3	A formal definition of Adequacy Criterion	50
Annex B (informative)	Guidance on the provisions of this standard	52
B.1	Scope	52
B.1.1	Purpose	52
B.1.2	Field of application	53
B.2	Normative references	53
B.3	Terms and definitions	53
B.4	General requirements	54
B.4.1	Quality management system	54
B.4.2	risk management	54
B.4.3	Software process rigor level classification	54
B.4.4	legacy software	56
B.5	Software development process	58
B.5.1	Software development planning	58
B.5.2	Software requirements analysis	58
B.5.3	Software architecture design	59
B.5.4	Software design	61
B.5.5	software unit implementation and verification	62
B.5.6	Software integration and integration testing	62
B.5.7	software system testing	63
B.5.8	Software release	64
B.6	Software maintenance process	64
B.6.1	Establish software maintenance plan	64
B.6.2	Problem and modification analysis	64
B.6.3	Modification implementation	65
B.7	Software risk management process	65
B.7.1	Analysis of software contributing to hazardous situations	66
B.8	Software configuration management process	66
B.8.1	Configuration identification	66
B.8.2	Change control	66
B.8.3	Configuration status accounting	67
B.9	Software problem resolution process	67
Annex C (informative)	Relationship to other standards	68
C.1	General	68
C.2	Relationship to ISO 13485	68
C.3	Relationship to ISO 14971	69
C.4	Relationship to PEMS requirements of IEC 60601-1:2005 + IEC 606011:2005/AMD1:2012	70
C.4.1	General	70
C.4.2	Software relationship to PEMS development	70
C.4.3	Development process	71
C.4.4	Maintenance process	71
C.4.5	Other process	71
C.4.6	Coverage of PEMS requirements in IEC 60601-1:2005 + IEC 606011:2005 /AMD1:2012	71
C.4.7	Relationship to requirements in IEC 60601-1-4	75

C.5	Relationship to IEC 61010-1	75
C.6	Relationship to ISO/IEC/IEEE 12207 (ISO/IEC/IEEE 12207 [13])	77
C.7	Relationship to IEC 61508	81
C.8	Relationship to IEC 60601-4-5 and IEC 81001-2-2.....	82
C.9	Relationship to IEC 81001-5-1	82
Annex D (informative)	Implementation	83
D.1	Introduction	83
D.2	Quality management system.....	83
D.3	evaluation quality management process	83
D.4	Integrating requirements of this standard into the manufacturer quality management process	83
D.5	Checklist for small manufacturer without a certified QMS.....	83
Annex E (informative)	Evolving Technology & Methodologies with focus on AI/ML....	85
E.1	Introduction	85
E.1.1	Evolving Technology & Methodologies	85
E.1.2	artificial intelligence (capability) in Healthcare	85
E.1.3	AI Performance evaluation Planning	87
E.1.4	AI Documentation.....	87
E.2	Supporting Requirements in the AI life-cycle.....	88
E.3	AI Bibliography	90
Annex F (informative)	Agile Development	92
F.1	Agile Development Approach.....	92
F.1.1	Overview	92
F.1.2	Planning for Development using Agile Approach	92
F.1.3	Agile Layers	93
F.1.4	Software Integration and Testing using Agile Approach	94
F.1.5	software system Testing using Agile Approach	94
F.1.6	Conclusion	94
Annex G (informative)	legacy software	95
G.1	risk management activity	95
G.2	Gap analysis	95
G.3	Gap closure activity	95
G.4	Rationale for use of legacy software	95
Bibliography		96
Figure 1	– Overview of software development processes and activities	10
Figure 2	– Overview of software maintenance processes and activities	10
Figure 3	– Software Development with additional AI-specific activities.....	11
Figure 4	– AI-enabled Device-specific Activities and Tasks included in Traditional V-Model	12
Figure 5	– Software Development Life cycle with Integration of AI Development Life cycle	13
Figure 6	– Assigning software process rigor (3.20) level	23
Figure B.1	– Pictorial representation of the relationship of hazard (3.10), sequence of events, hazardous situation (3.11), and harm (3.9) – from 14971:2007 Annex E (2019?)	55

Figure C.1 – Relationship of key medical device (3.15) (3.15) standards to IEC 62304	68
Figure C.2 – Software as part of the V-model	70
Figure C.3 – Application of IEC 62304 with IEC 61010-1	76
Figure E.1 – ELR (Ethics, Lawfulness & Robustness) Framework	86
Figure E.2 – AI Change Management (AICM) Framework for Trustworthy AI in Healthcare	86
Figure F.1 – Agile's Incremental / Evolutionary Life cycle	93
Table 1	8
Table A.1 – Summary of requirements by software process rigor (3.20) level	49
Table B.1 – Development (model) strategies as defined in ISO/IEC 12207	52
Table C.1 – Relationship to EN ISO 13485:2016	68
Table C.2 – Relationship to ISO 14971:2007	69
Table C.3 – Relationship to IEC 60601-1	71
Table C.4 – Relationship to [17]	77
Table D.1 – Checklist for small companies without a certified QMS	83

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**HEALTH SOFTWARE –
SOFTWARE LIFE CYCLE PROCESSES –****FOREWORD**

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC 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) IEC draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). IEC takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, IEC [had/had not] received notice of (a) patent(s), which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at <https://patents.iec.ch>. IEC shall not be held responsible for identifying any or all such patent rights.

IEC 62304 has been prepared by a maintenance team of subcommittee 62A: Common aspects of electrical equipment used in medical practice, of IEC technical committee 62: Electrical equipment in medical practice. It is an International Standard.

Note to National Committees

This is the first draft of a major revision and it has been edited using OSD.

The views generated from the current OSD version may distort defined terms or formatting - while the OSD content is still correct.

Authors are aware that spelling may be wrong and references to other standards may be wrong, obsolete or improperly formatted.

Please focus on technical specifications.

At this stage, MT49 may or may not consider editorial comments or hints regarding wrong referencing of standards.

Table C.4 was prepared by ISO/IEC JTC 1/SC 7, Software and **system** engineering.

It is published as an IEC standard.

Note to National Committees

The revision of this document follows the "62304 Design Specification" of 2023 which was developed by IEC SC62A (see SC62A/1456/DC) and agreed on by the National Committees (IEC SC62A/1470/INF). Any comments contradicting the Design Specification could only be addressed by a consensus procedure beyond MT49.

The new, larger scope is now Health Software as defined in Electropedia / IEC 60050-880.

As an effect of broadening the scope to Health Software (IEC CDV 60050-880), the processes for Risk Management and Quality System can not assumed by conforming to ISO 14971 and ISO 13485 but are assumed to be defined at the product-level.

For improved clarity, the number of safety-related categories was reduced from three to two: The former safety classes A,B,C were mapped to process rigor levels I or II in a systematic fashion.

The text of this International Standard is based on the following documents:

Table 1

IEC SC62A/1456/DC, IEC SC62A/1470/INF https://assets.iec.ch/public/sc62a/IEC%2062304%20Ed.%202.0%20Design%20Specification.pdf	Design Specification for Edition 2
IEC SC62A/1349/CDV IEC SC62A/1383A/RVC	Health Software - Lifecycle Processes
IEC 62304:Ed1.1:2015	Medical Software - Lifecycle Processes (published)

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above Table 1.

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/publications.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn, or
- revised.

NOTE 1 The attention of National Committees is drawn to the fact that equipment [manufacturer](#) and testing organizations may need a transitional period following publication of a new, amended or revised IEC or ISO publication in which to make products in accordance with the new requirements and to equip themselves for conducting new or revised tests. It is the recommendation of the committee that the content of this publication be adopted for mandatory implementation nationally not earlier than 3 years from the date of publication.

NOTE 2 The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

Software is often an integral part of **medical device** technology. Establishing the **safety** and effectiveness of a **medical device** containing software requires knowledge of what the software is intended to do and demonstration that the use of the software fulfils those intentions without causing any unacceptable **risk**.

The second edition intends to apply established software **process** elements to **health software** which is 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** (IEV).

Note to National Committees

As the new, larger scope may include non-medical software, implementation of ISO 14971 for Risk Management or ISO 13485 for a Quality System can not be assumed.

As a result, this document does neither normatively reference ISO 13485 nor ISO 14971.

This standard provides a framework of life cycle **process** with **activity** and tasks necessary for the safe design and maintenance of **health software**. This standard provides requirements for each life cycle **process**. Each life cycle **process** consists of a set of **activity**, with most **activity** consisting of a set of **task**.

As a basic foundation it is assumed that **health software** is developed and maintained within a product-level **process** that has quality management (see 4.1) and a **risk management** system (see 4.2). Some minor additional **risk management** requirements are needed for software, especially in the area of identification of contributing software factors related to **hazard**. These requirements are summarized and captured in Clause 7 as the software **risk management process**.

Whether software is a contributing factor to a **hazardous situation** is determined during the **hazard** identification **activity** of the **risk management process**. **hazardous situation** that could be indirectly caused by software (for example, by providing misleading information that could cause inappropriate treatment to be administered) need to be considered when determining whether software is a contributing factor. The decision to use software to control **risk** is made during the **risk control activity** of the **risk management process**.

The software development **process** consists of a number of **activity**. These **activity** are shown in Figure 1 and described in Clause 5. Because many incidents in the field are related to service or maintenance of **medical device system** including inappropriate software updates and upgrades, the software maintenance **process** is considered to be as important as the software development **process**. The software maintenance **process** is very similar to the software development **process**. It is shown in Figure 2 and described in Clause 6.

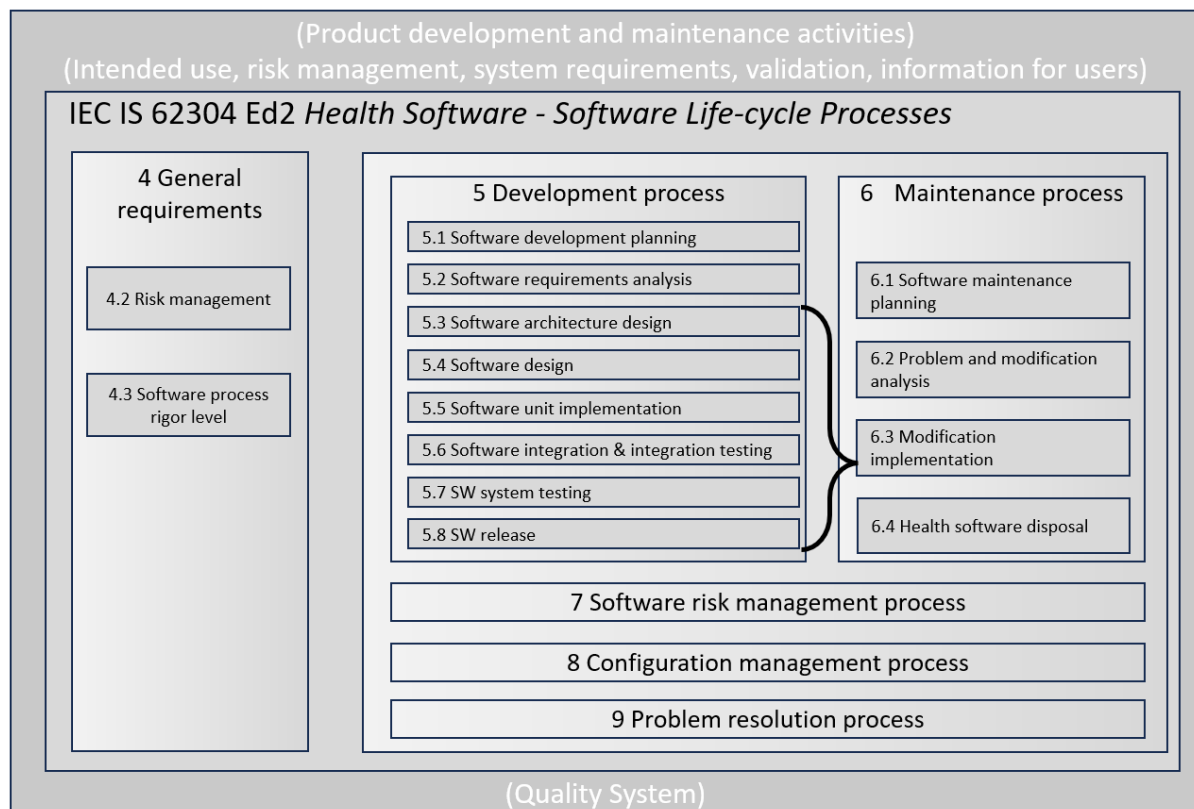


Figure 1 – Overview of software development processes and activities

This standard identifies two additional **process** considered essential for developing safe **health software**. They are the software configuration management **process** (Clause 8) and the software problem resolution **process** (Clause 9).

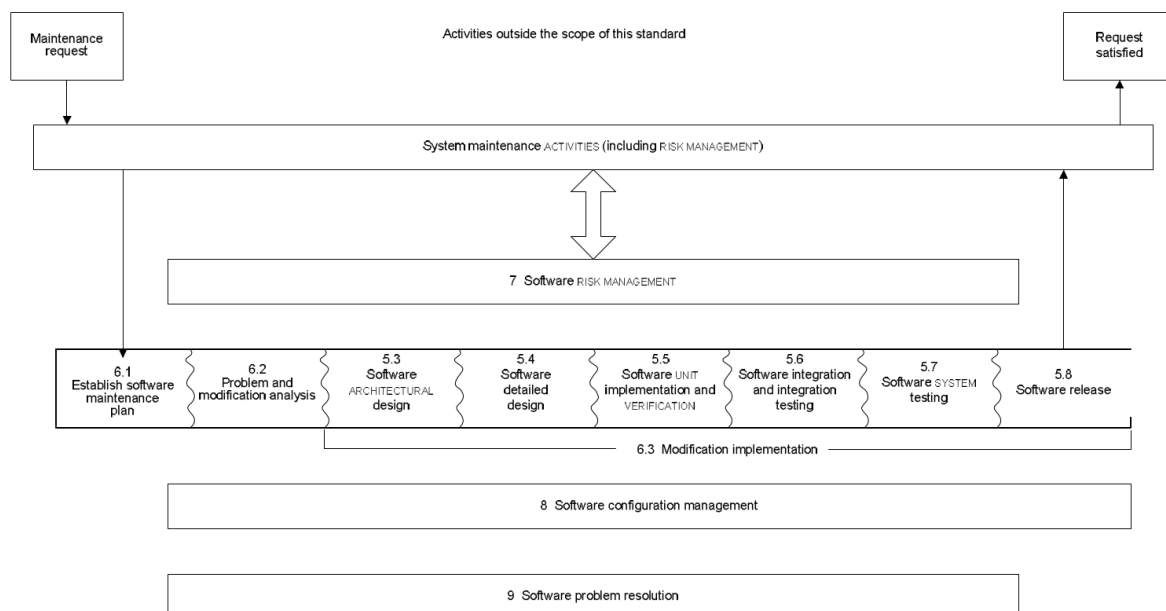


Figure 2 – Overview of software maintenance processes and activities

artificial intelligence (capability)

The software development processes are extended to capture additional activities and tasks when dealing with AI-enabled devices (AIeD) Figure 3.

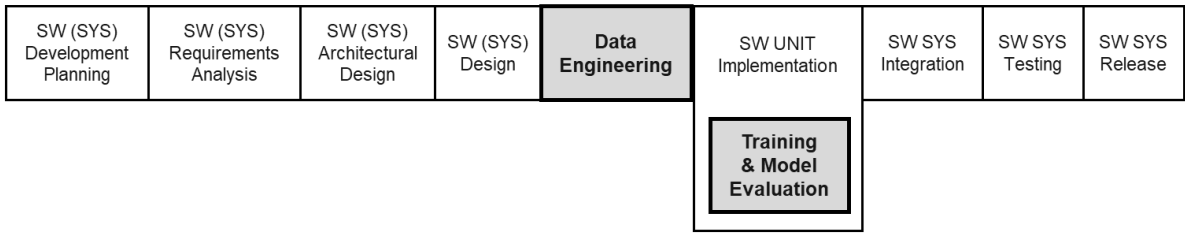


Figure 3 – Software Development with additional AI-specific activities

Note to National Committees:

In case, the function of some software item essentially depends on data obtained from training, that data shall be managed according to an AI Data Life-Cycle. That software item will then be called "AI-enabled".

IEC 62304 does neither specify methods for the data life-cycle nor does it require certain quality attributes other than what is covered by verification and testing in Clause 5.

As state-of-the-art for such data life-cycles is evolving, this document only calls for planning of those data life-cycle activities identified by the AI Framework presented by the FDA in summer 2024. Authors are aware that there are related IEC specifications under development, yet this document needs to refer to state-of-the-art.

The current thinking is that the entire data set obtained from training ("Model") will be used by the AI-enabled software item. Otherwise the software life-cycle is specified to be independent from the data life-cycle.

Annex E gives valuable insight into things that manufacturers need to consider in the data life-cycle. Authors are planning to update this Annex E and also update the figure showing how software life-cycle and data life-cycle are related.

Annex E provides additional guidance for the inclusion of [artificial intelligence \(capability\)](#) or Machine Learning (AI/ML) technologies to be integrated into the software development [process](#), herein after referred to as AI-enabled Devices (AIeD).

Additionally, Figure 4 provides an adapted V-Model as an example, with AI-specific tasks and activities highlighted. Whilst it is accepted that an agile approach is typical within current software development practices and would be particularly relevant for AI-enabled devices, the V-model continues to provide a clear structure, which is intended to support the identification of the necessary additional processes, tasks and activities that are required for traditional software development and AI-specific tasks and activities. The [manufacturer](#) defines the particular development methods and approach, taking account of the State of the Art (SoTA) and accepted development engineering practices, whilst ensure the relevant requirements are met, ref. Annex F.

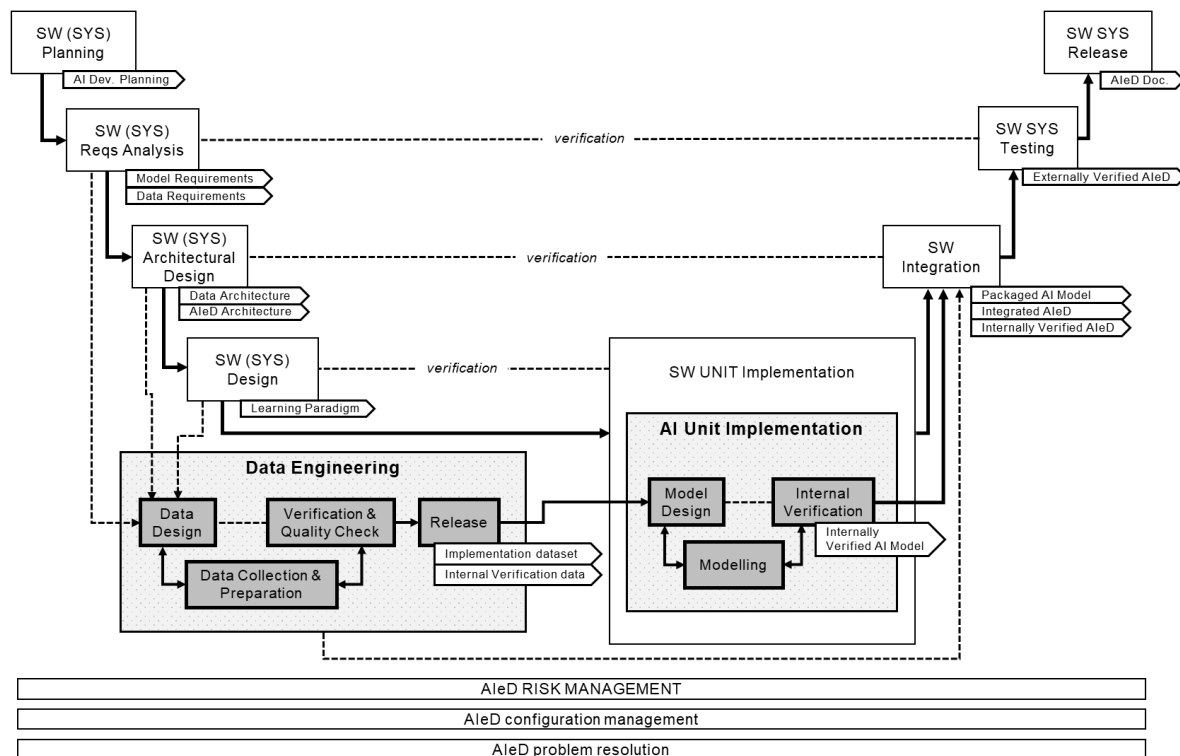


Figure 4 – AI-enabled Device-specific Activities and Tasks included in Traditional V-Model

NOTE Legend for Figure 4

- White box: development activities
- Light grey: AI-specific activities
- White rectangles: AI-related outcomes
- Solid lines: direction of flow
- Dash lines: optional direction

health software development integrates the AI data life-cycle through deliverables, which is why all additional AI requirements have been added to the planning phase (5.1.15) - to ensure that the manufacturer has adequately planned and integrated the AI development process into the existing Software Development Processes, Tasks and Activities. During planning, the manufacturer identifies the AI-specific tasks necessary throughout the Software / AI development life-cycle(s). Key tasks and activities for each phase are identified, ref. Figure 5.

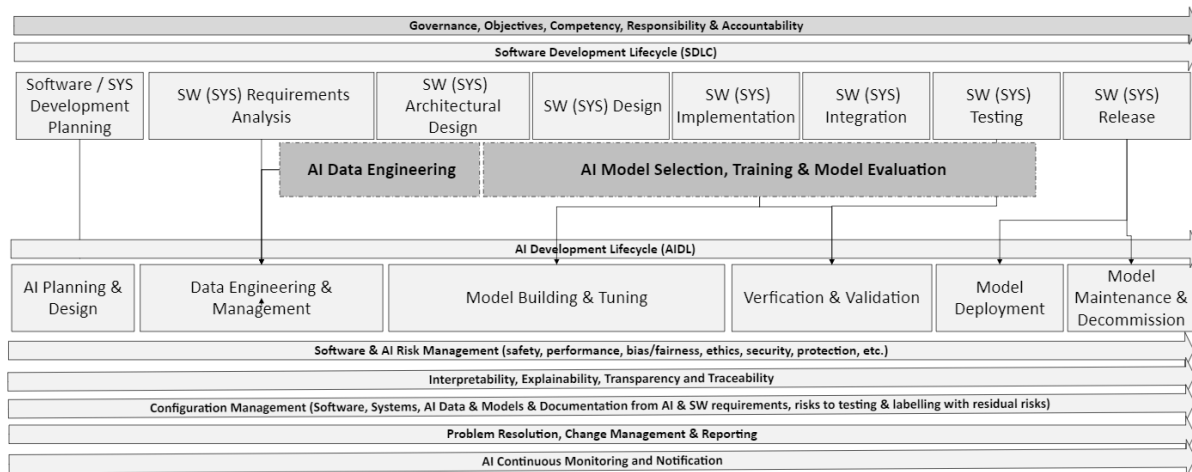


Figure 5 – Software Development Life cycle with Integration of AI Development Life cycle

legacy software

Annex G updates the standard to add requirements to deal with **legacy software**, where the software design is prior to the existence of the current version of this standard, to assist manufacturers who wish to show regulatory compliance by using state-of-the-art methods in software development. Software **process rigor** level changes include clarification of requirements and updating of the software **process rigor** level to include a **risk**-based approach.

Structure

This standard does not specify an organizational structure for the **manufacturer** or which part of the organization is to perform which **process**, **activity**, or **task**. This standard requires only that the **process**, **activity**, or **task** be completed to establish conformance with this standard.

Documentation

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

Life cycle Model

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

Rationale & Provisions

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

For the purposes of this standard:

- "shall" means that conformance to a requirement is mandatory for conformance to this standard;
- "should" means that conformance to a requirement is recommended but is not mandatory for conformance to this standard;
- "may" is used to describe a permissible way to achieve conformance to a requirement;
- "establish" means to define, document, and implement; and

- where this standard uses the term "as appropriate" in conjunction with a required [process](#), [activity](#), [task](#) or output, the intention is that the [manufacturer](#) shall use the [process](#), [activity](#), [task](#) or output unless the [manufacturer](#) can document a justification for not so doing.

HEALTH SOFTWARE – SOFTWARE LIFE CYCLE PROCESSES –

1 Scope

1.1 Purpose

NOTE Guidance for this clause can be found in Annex B.

This standard defines the life cycle requirements for **health software**. The set of **process**, **activity**, and **task** described in this standard establishes a common framework for **health software** life cycle **process**.

1.2 Field of application

NOTE 1 Guidance for this clause can be found in Annex B.

This standard 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 intended to be used specifically for managing, maintaining, or improving health of individuals, or for the delivery of care;
- **health software** with **artificial intelligence (capability)** (AI) or machine learning (ML) algorithms

NOTE 2 Examples of **health software** include the following:

- a) software as a part of a **medical device**: software that is an integral part of a device such as an infusion pump or dialysis machine.
- b) software as part of specific health hardware: patient diagnostic wristband, printer software, healthcare scanner software, diagnostic app on specific wearable hardware (i.e. watch, wristband, chestband).
- c) software as a **medical device** (SaMD): software that is itself a **medical device**, such as a software application that performs diagnostic image analysis for making treatment decisions. A definition of software as a **medical device** is provided in [44] 1.
- d) software-only product for other health use: hospital information systems, electronic health records, electronic medical records, mobile diagnostic applications running on devices without physiologic sensors or detectors, clinical software as a service, i.e. software executed in an external environment, providing diagnostic calculation-results that fulfil the definition of a **medical device**.

Before any type of software can be placed into service, activities are necessary before the software is integrated into the PRODUCT. These PRODUCT-LEVEL activities are not covered by this document (see Figure 1) but can be found in related product standards.

NOTE 3 This standard can be used in the development and maintenance of software that either is embedded into a **medical device** or that is itself a **medical device**. In both cases, additional development activities are needed at the **system** level before this type of software can be placed into service. These **system** activities are not covered by this standard, but can be found in IEC 60601-1, IEC 82304-1¹ [1].

¹ In preparation.

In this document, chapters relate to **process**, sections relate to **activity**, and **task** relate to subsections.

This document describes **process** that are intended to be applied to software which executes on a processor or which is executed by other software (for example an interpreter) which executes on a processor.

This document applies regardless of the persistent storage device(s) used to store the software (for example: hard disk, optical disk, permanent or flash memory).

This document applies regardless of the method of delivery of the software (for example: transmission by network or email, optical disk, flash memory or EEPROM). The method of software delivery itself is not considered **health software**.

This document does not cover the means of validation of the final product, even when the product consists entirely of software. It also does not cover software life cycle steps after **release** for intended use of the product, including implementation, configuration, integration (with other systems), go-live, clinical use, operations, decommissioning or disposal, other than **activity** involving maintenance of the software.

This document does not cover the validation of software tools used in the design of medical devices (e.g. computer aided design (CAD) software), software used in **medical device** quality systems or software for regulated processes (see ISO/TR 80002-2 [20] and AAMI TIR 36 [40]).

Data quality and validation of emergent characteristics or functionality of **artificial intelligence (capability)** (AI) **health software** are not within the scope of this document.

NOTE 4 The 2nd edition of this standard recognizes that there is a data life-cycle for training data used as an essential input to AI-enabled software items. Users of this standard may need to utilize other standards and technical sources to supplement this document in addressing the unique performance characteristics of their AI.

NOTE 5 If a product incorporates embedded software intended to be executed on a processor, the requirements of this document apply to the software, including the requirements concerning SOFTWARE OF UNKNOWN PROVENANCE (SOUP) – see 8.1.2)

NOTE 6 Validation and other development activities are needed at the product level before the software and incorporating products can be placed into service. These product-level activities are not covered by this standard but can be found in related product standards (e.g. IEC 60601-1, IEC 82304-1, etc.).

1.3 Relationship to other standards

This **health software** life-cycle document is written in a way that it can be used together with referencing standards when developing and maintaining a product that includes **health software** (see Annex C).

1.4 Conformance

Conformance is determined by inspection of all the documentation required by this document including an assessment of processes, tasks and activities required for the software **process rigor** level.

Conformity of **legacy software** may be demonstrated as indicated in Annex G

2 Normative references

There are no normative references in this document.

3 Terms and definitions

NOTE Guidance for this clause can be found in Annex A.

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

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

– IEC Electropedia: available at www.electropedia.org/

– ISO Online browsing platform: available at www.iso.org/obp

3.1

activity

set of one or more interrelated or interacting [task](#)

3.2

anomaly

condition that deviates from the expected based on requirements specifications, design documents, standards, etc. or from someone's perceptions or experiences. Anomalies may be found during, but not limited to, the review, test, analysis, compilation, or use of [medical device software](#) or applicable documentation

Note 1 to entry: Based on IEEE 1044:1993, definition 3.1.

3.3

architecture

organizational structure of a [system](#) or component

[SOURCE: IEEE 610.12:1990]

3.4

artificial intelligence (capability)

capability of a [software item](#) to perform functions that are generally associated with human intelligence, such as reasoning and learning

Note 1 to entry: Modified from IEC 60050 176-09-17, uses [software item](#) instead of functional unit

Note 2 to entry: .Refer to ISO/IEC 22989:2023 for AI Terms and Definitions.

Note 3 to entry: This document uses artificial intelligence in the context of functions essentially depending on data obtained from training.

3.5

change request

documented specification of a change to be made to a [medical device software](#)

3.6

configuration item

entity that can be uniquely identified at a given reference point

Note 1 to entry: Based on ISO/IEC 12207:2008, 4,7.

3.7

deliverable

required result or output (includes documentation) of an [activity](#) or [task](#)

3.8

evaluation

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

[SOURCE: ISO/IEC 12207:2008, 4.12]

3.9

harm

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

[SOURCE: ISO 14971:2007, 2.2]

3.10

hazard

potential source of [harm](#)

[SOURCE: ISO 14971:2007, 2.3]

3.11

hazardous situation

circumstance in which people, property or the environment are exposed to one or more hazard

[SOURCE: ISO 14971:2007, 2.4]

3.12

health software

software:

- intended to be used specifically for managing, maintaining, or improving health (IEV 880-10-05) of individuals, or for the delivery of care;
- developed for the purpose of being incorporated into a [medical device](#) (IEV 880-09-24); or
- developed for the purpose of being incorporated into an MES (IEV 880-09-27)

Note 1 to entry: [health software](#) fully includes [medical device software](#).

Note 2 to entry: Software incorporated into an MES but not developed for that purpose (e.g. commercial off the shelf software or COTS software) is not considered [health software](#).

[SOURCE: IEC 880-20-07]

3.13

legacy software

[health software](#) previously developed according to the generally recognized *state of the art* (IEV 880-10-13) at the time of design

Note 1 to entry: Adequate *records* exist that demonstrate *conformity* with *state of the art* at the time of design (e.g. an earlier *version* of a standard).

Note 2 to entry: *Life cycle processes* typically cannot be repeated in their entirety to demonstrate *conformity* with current *state of the art*.

[SOURCE: Based on IEC 880-10-06 legacy, adj.]

3.14

manufacturer

organization (IEV 880-13-15) with responsibility for design or manufacture of [health software](#) with the intention of making it available for use, under their name;

[SOURCE: IEC 60050:880-13-14, broadened to fit the scope of this document]

Note 1 to entry: Attention is drawn to the fact that the provisions of national or regional regulations can apply to the definition of [manufacturer](#).

Note 2 to entry: For a definition of labelling, see ISO 13485:2003, definition 3.6.

[SOURCE: ISO 14971:2007, 2,8]

3.15

medical device

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

- diagnosis, prevention, monitoring, treatment or alleviation of disease,
- diagnosis, monitoring, treatment, alleviation of or compensation for an injury,
- investigation, replacement, modification, or support of the anatomy or of a physiological [process](#),
- supporting or sustaining life,
- control of conception,
- disinfection of [medical device](#),

- providing information for medical purposes by means of in vitro examination of specimens derived from the human body,

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

Note 1 to entry: This definition has been developed by the Global Harmonization Task Force (GHTF). See bibliographic reference ISO/IEC 90003:2014 [2] (in ISO 13485:2003).

Note 2 to entry: Some differences can occur in the definitions used in regulations of each country.

Note 3 to entry: In conjunction with IEC 60601-1:2005 and IEC 60601-1:2005/AMD1:2012 the term "[medical device](#)" assumes the same meaning as the equipment or the [system](#) (which are defined terms of IEC 60601-1).

[SOURCE: ISO 13485:2003, definition 3.7]

3.16

medical device software

[software system](#) that has been developed for the purpose of being incorporated into the [medical device](#) being developed or that is intended for use as a [medical device](#)

Note 1 to entry: This includes a [medical device](#) software product, which then is a [medical device](#) in its own right.

3.17

negligible harm

[harm](#) resulting in inconvenience or temporary discomfort

Note 1 to entry: Typically, the lowest category of severity of [harm](#) defined in the [risk management process](#) used on product level

3.18

PEMS

programmable electrical medical systems

3.19

problem report

record of actual or potential behaviour of a [medical device software](#) that a user or other interested person believes to be unsafe, inappropriate for the intended use or contrary to specification

Note 1 to entry: This standard does not require that every [problem report](#) results in a change to the [medical device software](#). A [manufacturer](#) can reject a [problem report](#) as a misunderstanding, error or insignificant event.

Note 2 to entry: A [problem report](#) can relate to a released [medical device software](#) or to a [medical device software](#) that is still under development.

Note 3 to entry: This standard requires the [manufacturer](#) to perform extra decision making steps (see Clause 6) for a [problem report](#) relating to a released product to ensure that regulatory actions are identified and implemented.

3.20

process rigor

stringent enforcement and conformance to given formal and structured processes

3.21

process

set of interrelated or interacting [activity](#) that transform inputs into outputs

Note 1 to entry: The term "[activity](#)" covers use of resources.

[SOURCE: ISO 9000:2000, definition 3.4.1]

3.22

regression testing

testing required to determine that a change to a [system](#) component has not adversely affected functionality, reliability or performance and has not introduced additional defects

[SOURCE: ISO/IEC 90003:2004, definition 3.11]

3.23**release**

particular [version](#) of a [configuration item](#) that is made available for a specific purpose

Note 1 to entry: Based on ISO/IEC 12207:2008, definition 4.35.

3.24**residual risk**

[risk](#) remaining after [risk control](#) measures have been taken

Note 1 to entry: Adapted from ISO/IEC Guide 51:1999, definition 3.9.

Note 2 to entry: ISO/IEC Guide 51:1999, definition 3.9 uses the term "protective measures" rather than "[risk control](#) measures." However, in the context of this International Standard, "protective measures" are only one option for controlling [risk](#) as described in 6.2 [of ISO 14971:2007].

[SOURCE: ISO 14971:2007, 2.15.]

3.25**risk**

combination of the probability of occurrence of [harm](#) and the severity of that [harm](#)

[SOURCE: ISO 14971:2007, 2.16]

3.26**risk analysis**

systematic use of available information to identify [hazard](#) and to estimate the [risk](#)

[SOURCE: ISO 14971:2007, 2.17]

3.27**risk control**

[process](#) in which decisions are made and [risk](#) are reduced to, or maintained within, specified levels

[SOURCE: ISO 14971:2007, 2.19]

3.28**risk estimation**

[process](#) used to assign values to the probability of occurrence of [harm](#) and the severity of that [harm](#)

[SOURCE: ISO 14971:2007 2.20]

3.29**risk evaluation**

[process](#) of comparing the estimated [risk](#) against given [risk](#) criteria to determine the acceptability of the [risk](#)

[SOURCE: ISO 14971:2007 2.21]

3.30**risk management**

systematic application of management policies, procedures, and practices to the [task](#) of analysing, evaluating, and controlling [risk](#)

[SOURCE: ISO 14971:2007, 2.22]

NOTE modified – The phrase "and monitoring" has been removed

3.31**risk management file**

set of records and other documents, not necessarily contiguous, that are produced by a [risk management process](#)

[SOURCE: ISO 14971:2007, 2.23]

3.32 safety

freedom from unacceptable [risk](#)

[SOURCE: ISO 14971:2007, 2.24]

3.33 security cybersecurity

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 violation of confidentiality, integrity, and availability are maintained at an acceptable level throughout the life-cycle

Note 1 to entry: SOURCE: ISO 81001-1:2021 [17], 3.2.13

3.34 software development life cycle model

conceptual structure spanning the life of the software from definition of its requirements to its [release](#), which:

- identifies the [process](#), [activity](#) and [task](#) involved in development of software,
- describes the sequence of and dependency between [activity](#) and [task](#), and
- identifies the milestones at which the completeness of specified [deliverable](#) is verified.

Note 1 to entry: Based on ISO/IEC 12207:1995, definition 3.11

3.35 software item

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

Note 1 to entry: Three terms identify the software decomposition. The top level is the [software system](#). The lowest level that is not further decomposed is the [software unit](#). All levels of composition, including the top and bottom levels, can be called [software item](#). A [software system](#), then, is composed of one or more [software item](#), and each [software item](#) is composed of one or more [software unit](#) or decomposable [software item](#). The responsibility is left to the [manufacturer](#) to provide the granularity of the [software item](#) and [software unit](#).

Note 2 to entry: Based on ISO/IEC 90003:2004, 3.14 and ISO/IEC 12207:2008, 4.41

3.36 software system

integrated collection of [software item](#) organized to accomplish a specific function or set of functions

3.37 software unit

[software item](#) that is not subdivided into other items

Note 1 to entry: The granularity of [software unit](#) is defined by the [manufacturer](#) (see B.3).

3.38 soup software of unknown provenance (acronym)

[software item](#) that is already developed and generally available and that has not been developed for the purpose of being incorporated into the [medical device](#) (also known as “off-the-shelf software”) or [software item](#) previously developed for which adequate records of the development [process](#) are not available

Note 1 to entry: A [medical device software system](#) in itself cannot be claimed to be [soup](#).

3.39 system

integrated composite consisting of one or more of the [process](#), hardware, software, facilities, and people, that provides a capability to satisfy a stated need or objective

Note 1 to entry: Based on ISO/IEC ISO/IEC 12207:2008, 4.48.

3.40

task

single piece of work that needs to be done

3.41

traceability

degree to which a relationship can be established between two or more products of the development [process](#)

Note 1 to entry: Requirements, [architecture](#), [risk](#) control measures, etc. are examples of deliverables of the development [process](#).

[SOURCE: IEEE 610.12:1990]

3.42

verification

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

Note 1 to entry: "Verified" is used to designate the corresponding status.

Note 2 to entry: In design and development, [verification](#) concerns the [process](#) of examining the result of a given [activity](#) to determine conformity with the stated requirement for that [activity](#)..

[SOURCE: ISO 9000:2000, definition 3.8.4]

3.43

version

identified instance of a [configuration item](#)

Note 1 to entry: Modification to a [version](#) of [medical device software](#), resulting in a new [version](#), requires software configuration management action.

Note 2 to entry: Based on ISO/IEC 12207:2008, 4.56.

4 General requirements

NOTE Guidance for this clause can be found in Annex B.

4.1 Not used

4.2 [risk management](#)

NOTE 1 Guidance for this clause can be found in Annex B.

For the purposes of this document, the product's [risk management process](#) is used for the [health software](#) life cycle activities and tasks.

NOTE 2 The [risk analysis](#) from the product [risk management](#) is being used 4.3 in order to determine [process rigor](#) level.

NOTE 3 Since this document is not a product level standard, any [risk management](#) activities of the [health software](#) life-cycle [process](#) can be derived from the manufacturers' [risk management](#) system applied to their products.

NOTE 4 The product security [risk management](#) informs the product [risk management process](#) and vice versa.

NOTE 5 See IEC 81001-5-1:2021, section 4.2

NOTE 6 The information flow between software development and the product life-cycle is specified in Clause 7, which specifies how the [risk management process](#) assesses and evaluates the contribution of software to hazardous situations.

NOTE 7 The [risk management process](#) used at the product level determines the rigor of the software [process](#).

NOTE 8 Applicable jurisdictional requirements call out application of ISO 14971.

4.3 [Software process rigor level](#)

4.3.1 [software system](#)

- a) The [manufacturer](#) shall assign to each [software system](#) a software [process rigor](#) level (I or II) according to the [risk](#) of [harm](#) to which the [software system](#) can contribute as indicated in Figure 6

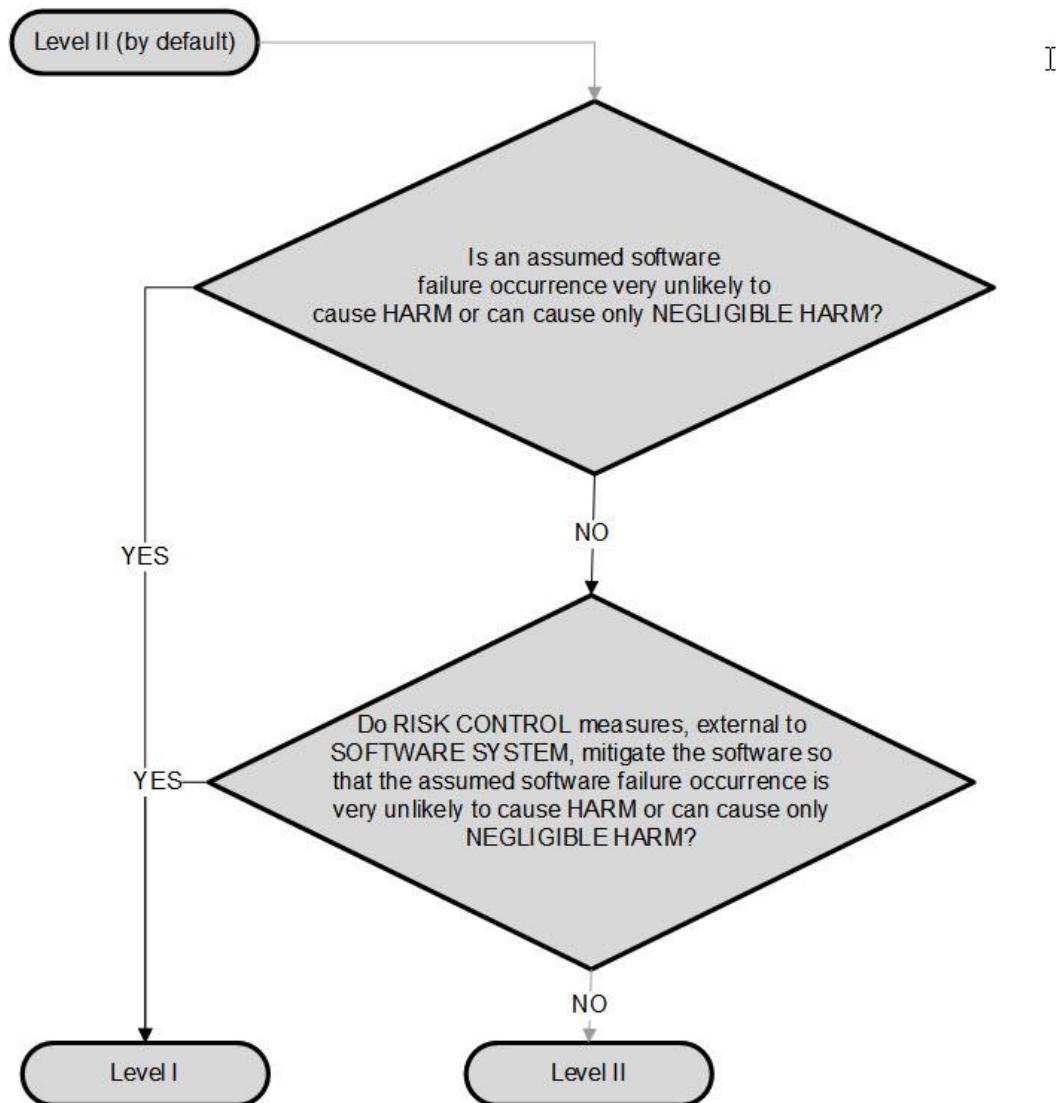


Figure 6 – Assigning software process rigor (3.20) level

When determining the software **process rigor** level, the entire software functionality with all possible failures shall be evaluated. No probability shall be assigned to the occurrence of software failures, instead a worst-case scenario shall be assumed.

The **software system** is software **process rigor** level I if:

- An assumed software failure occurrence is very unlikely to cause **harm** or can cause only **negligible harm**; or
- **risk control** measures, external to **software system**, mitigate the software so that the assumed software failure occurrence is very unlikely to cause **harm** or can cause only **negligible harm**.

The **software system** is software **process rigor** level II if the level I conditions are not met.

The **risk control** measures that reduce the software **process rigor** level shall be **verification** for effectiveness prior to software **release**.

- b) The **manufacturer** shall document the software **process rigor** level assigned to each **software system** in the **risk management file**.

- c) For each **software system** until a software **process rigor** level is assigned, all requirements shall apply.

NOTE 1 For a **software system** that initially requires software **process rigor** level II, the **manufacturer** may implement additional **risk control** measures external to the **software system** (including revising the **system architecture** containing the **software system**) and subsequently assign a new software **process rigor** level to the **software system**.

NOTE 2 External **risk control** measures can be realized by hardware, an independent **software system**, or health care procedures.

NOTE 3 **verification** of **risk control** measures can be met by using a **risk management process** that is compliant with ISO 14971.

4.3.2 Decomposition into **software item**

- a) When a **software system** with software **process rigor** level II is decomposed into **software item**, and when a **software item** is decomposed into further **software item**, such **software item** shall inherit the software **process rigor** level of the original **software item** (or **software system**) unless the **manufacturer** documents a rationale for the assignment of software **process rigor** level I according to 4.3.1 a) replacing **software system** with **software item**, and the **software item** is effectively segregated from **software item** with software **process rigor** level II.

NOTE 1 Decomposition is the static refinement of a **software system** performed as part of the software architectural design (5.3) and the software design (5.3.6)

NOTE 2 See B.4.3 for further guidance

- b) **software item** that implement a **risk control** measure shall be assigned a software **process rigor** level based on the **risk** that the **risk control** measure is controlling.

5 Software development **process**

5.1 Software development planning

NOTE Guidance for this clause can be found in Annex B.

5.1.1 Software development plan

The **manufacturer** shall establish a software development plan (or plans) for conducting the **activity** of the software development **process** appropriate to the scope, magnitude, and software **process rigor** level of the **software system** to be developed. The **software development life cycle model** shall either be fully defined or be referenced in the plan (or plans). The plan shall address the following:

- a) the **process** to be used in the development of the **software system** (see **Note 4**);
- b) the **deliverable** (includes documentation) of the **activity** and **task**;
- c) **traceability** between **system** requirements, software requirements, **software system** test, and **risk control** measures implemented in software;
- d) software configuration and change management, including **software configuration item** and software used to support development; and
- e) software problem resolution for handling problems detected in the **health software**, **deliverable** and **activity** at each stage of the life cycle.

[Levels I, II]

NOTE 1 The **software development life cycle model** can identify different elements (**process**, **activity**, **task** and **deliverable**) for different **software item** according to the **process rigor** level classification of each **software item** of the **software system**.

NOTE 2 These **activity** and **task** can overlap or interact and can be performed iteratively or recursively. It is not the intent to imply that a specific life cycle model is preferred.

NOTE 3 Other **process** are described in this standard separately from the development **process**. This does not imply that they must be implemented as separate **activity** and **task**. The **activity** and **task** of the other **process** can be integrated into the development **process**.

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

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

NOTE 6 The Software development plan is applicable to all software [process rigor](#) levels with justification where a section might not be applicable.

5.1.2 Keep software development plan updated

The [manufacturer](#) shall update the plan as development proceeds as appropriate.

[Levels I, II]

5.1.3 Software development plan reference to [system](#) design and development

- a) As inputs for software development, [system](#) requirements shall be referenced in the software development plan by the [manufacturer](#).
- b) In the software development plan, the [manufacturer](#) shall include or reference procedures for coordinating the software development with the product development necessary to satisfy 4.1 (such as [system](#) integration, [verification](#), and validation).

[Levels I, II]

NOTE There might not be a difference between [software system](#) requirements and [system](#) requirements if the [software system](#) is a stand alone [system](#) (software-only device).

5.1.4 Software development standards, methods and tools planning

- a) The [manufacturer](#) shall include or reference in the software development plan:

- 1) standards,
- 2) methods, and
- 3) tools

associated with the development of [software item](#)

NOTE In the software development plan, "method" includes any technique to support software development, like programming languages, programming environments, paradigms, software modelling techniques, software development frameworks etc.

- b) The [manufacturer](#) shall document the methods used for the following activities, as required by the [process rigor](#) level:

- requirements [verification](#),
- [architecture verification](#),
- design [verification](#),
- unit [verification](#),
- integration testing, and
- [software system verification](#).

[Levels I, II]

5.1.5 Software integration and integration testing planning

The [manufacturer](#) shall include or reference in the software development plan, a plan to integrate the [software item](#) (including [soup](#)) and perform testing during integration. [Level II]

NOTE 1 It is acceptable to combine integration testing and [software system](#) testing into a single plan and set of [activity](#).

NOTE 2 See 5.6.

5.1.6 Software [verification](#) planning

The [manufacturer](#) shall include or reference in the software development plan the following [verification](#) information:

- a) [deliverable](#) requiring [verification](#);
- b) the required [verification task](#) for each life cycle [activity](#);
- c) milestones at which the [deliverable](#) are verified ([verification](#);) and
- d) the acceptance criteria for [verification](#) of the [deliverable](#).

[Levels I, II]

5.1.7 Software risk management planning

The **manufacturer** shall include or reference in the software development plan, a plan to conduct the **activity** and **task** of the software **risk management process**, including the management of **risk** relating to **soup** and security risks.

[Level II]

NOTE See Clause 7.

5.1.8 Documentation planning

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

- a) title, name or naming convention;
- b) purpose;
- c) procedures and responsibilities for development, review, approval and modification.

[Levels I, II]

NOTE See Clause 8 for consideration of configuration management of documentation.

5.1.9 Software configuration management planning

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

- a) the classes, types, categories or lists of items to be controlled;
- b) the software configuration management **activity** and **task**;
- c) the organization(s) responsible for performing software configuration management **activity**;
- d) their relationship with other organizations, such as software development or maintenance;
- e) when the items are to be placed under configuration control; and
- f) when the problem resolution **process** is to be used.

[Levels I, II]

NOTE See Clause 8.

5.1.10 Supporting items to be controlled

The **manufacturer** shall control items that could impact the **health software** development or are required to recreate a specific software **release**. This may include:

- Tools,
- Development environment,
- Settings,
- Data and,
- Simulators.

[Levels I,II]

NOTE 1 Examples of such items include compiler/assembler versions, make files, batch files, and specific environment settings.

NOTE 2 See Clause 8.

NOTE 3 Validation of non-product software is addressed by the quality system at product level.

5.1.11 Software configuration item control before verification

The **manufacturer** shall plan to place **configuration item** under configuration management control before they are **verification**.

[Levels I,II]

5.1.12 Not used

See 7.1.2

5.1.13 Software release planning

The **manufacturer** shall define or reference the procedure for software **release**, according to the purpose of the **release**, in the software development plan.

NOTE A software **release**, following this standard is not a product **release**. Additional **release** activities are expected at product level, for instance design validation and product documentation. This applies to all types of software including stand alone software.

[Levels I, II]

5.1.14 Software maintenance planning

The **manufacturer** shall include or reference a software maintenance plan (see 6.1) in the Software Development Plan

[Levels I,II]

5.1.15 artificial intelligence (capability) (AI) planning

This clause specifies AI-related deliverables for software items that rely upon **artificial intelligence (capability)** including Machine Learning (ML), Deep Learning (DL), Reinforcement Learning (RL) or Large Language Models (LLM), also referred to as AI-enabled Devices (AIEDs) or **health software** that depends on data obtained from an AIED.

The **manufacturer** shall identify the software items or functions that are AI-enabled (as above).

The **manufacturer** shall identify any general-purpose AI platform systems which are only intended to implement general functions, as SOUP.

NOTE 1 This clause assumes that AI-enabled clinical functions are executed on generic AI platform systems that are general-purpose software - independent of the product function - before being 'programmed' (e.g. configured, parametrized, and trained) towards specific health-related functionality. Such generic platforms are considered SOUP and managed using configuration management. See Clause 8.

For each AI-enabled **software item** or function the **manufacturer** shall plan the following activities and specify the methods and standards applied:

a) AIED Planning and Design

NOTE 2 This can include:

- 1) Governance and objectives
- 2) Resourcing and competency
- 3) Transparency and **traceability**
- 4) Data collection and quality planning
- 5) Ethics and fairness
- 6) **risk management** for **safety**, performance and fundamental rights
- 7) Security and data protection
- 8) Algorithm selection and model design
- 9) Feature engineering
- 10) Scalability, infrastructure and observability
- 11) Interpretability and explainability

b) Data collection and management

NOTE 3 This can include:

- 1) Data Suitability
- 2) Data quality and quantity
- 3) Data privacy and security
- 4) Data governance and configuration management
- 5) Data acquisition, augmentation and integration
- 6) Data labelling and annotation
- 7) Data sampling and bias mitigation
- 8) Data versioning and [traceability](#)
- 9) Data storage and infrastructure
- 10) Data ownership, access and sharing

Ref. IEC Publicly Available Specification (PAS) 62531 Data Management and Data Quality Assessment for Medical devices

c) Model-building and tuning

NOTE 4 This can include:

- 1) Model selection
- 2) Model analysis, selection and complexity
- 3) Feature selection and data engineering
- 4) Parameter / Hyper-parameter tuning
- 5) Cross-validation selection and design
- 6) Ensemble methods
- 7) Regularisation and optimization
- 8) Model explainability and interpretability
- 9) Model performance and [evaluation](#) metrics
- 10) Model robustness and generalization

d) [verification](#) and validation

NOTE 5 This can include:

- 1) [verification](#) and validation strategy
- 2) Performance [evaluation](#) metrics
- 3) Internal validation (model and data [verification](#))
- 4) External validation (clinical testing to user needs)
- 5) Explainability method [verification](#) and validation
- 6) Transparency, usability and interpretability
- 7) Technical documentation, [traceability](#) and reporting
- 8) Independent test strategy and competency

Ref. IEC 63450 AI Verification and Validation Test Methods

e) Model deployment

NOTE 6 This can include:

- 1) Scalability
- 2) Performance
- 3) Reliability
- 4) Usability

- 5) Integration
 - 6) Versioning
 - 7) Monitoring and logging
 - 8) Compliance and governance
 - 9) Documentation and training
 - 10) Deployment and decommission planning
- Ref. IEC 63521 AI Performance Evaluation

f) Operation and monitoring

NOTE 7 This can include:

- 1) Real-time monitoring
- 2) Alerting and notifications
- 3) Logging and auditing
- 4) Performance optimization
- 5) Scalability and resource management
- 6) Feedback mechanisms
- 7) Security and compliance
- 8) Model drift and bias detection
- 9) Labelling and instructions
- 10) Incident response and troubleshooting
- 11) Continuous improvement

g) Real-world performance [evaluation](#)

NOTE 8 This can include:

- 1) Define key performance indicators (KPIs)
- 2) Production data collection
- 3) [evaluation](#) metrics calculation
- 4) Comparison with baseline
- 5) Monitoring and alerting
- 6) Clinical oversight
- 7) Drift detection
- 8) Feedback collection
- 9) Error analysis
- 10) Continuous improvement
- 11) Change management
- 12) Re-training
- 13) Responsibility agreement
- 14) Documentation and reporting

Also see Annex E.

NOTE 9 A software life-cycle is detached from the data life-cycle. [release](#) of [health software](#) can be independent from changes related to some training [activity](#) like e.g. PCCP

[Levels I, II]

5.1.16 Communication plan

The **manufacturer** shall define or reference a communication strategy in the software development plan / AI development plan to inform the "users", "owners", "administrators", "regulators" about:

- a) any problem in the **health software** released for Intended use and the consequences of continued unchanged use; and
- b) the nature of any available changes to **health software** released for Intended use to obtain and install the changes.

NOTE As an example, the communication plan can refer to the product maintenance plan which then resolves the roles of "users", "owners", "administrators", "regulators" as they are used in Clause 6.

[Levels I,II]

5.2 Software requirements analysis

5.2.1 Define and document software requirements from **system** requirements

The **manufacturer** shall define and document **health software** requirements derived from the **system** level requirements.

[Levels I, II]

NOTE 1 **health software** requirements often refine **system** level requirements by including 'technical' specifications based on more general requirements at **system** level.

NOTE 2 There might not be a difference between **software system** requirements and **system** requirements if the **software system** is a stand alone **system** (software-only device).

5.2.2 Software requirements content

The **manufacturer** shall include at least the following types of software requirements as appropriate to the **health software**:

- a) functional and capability requirements;

NOTE 1 Examples include:

- performance (e.g., purpose of software, timing requirements),
- physical characteristics (e.g., code language, platform, operating system),
- computing environment (e.g., hardware, memory size, processing unit, time zone, network infrastructure) under which the software is to perform, and
- need for compatibility with upgrades or multiple **software** or other device versions.
- user-controlled functions supporting migration or disposal of **health software**

- b) **software system** inputs and outputs;

NOTE 2 Examples include:

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

- c) interfaces between the **software system** and other **system**;

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

- e) **security** requirements;

NOTE 3 Examples include:

- those related to the compromise of sensitive information,
- authentication,
- authorization,
- audit trail, and