

---

---

**Medical device software —**  
**Part 2:**  
**Validation of software for medical**  
**device quality systems**

*Logiciels de dispositifs médicaux —*

*Partie 2: Validation des logiciels pour les systèmes de qualité des*  
*dispositifs médicaux*



Reference number  
ISO/TR 80002-2:2017(E)



**COPYRIGHT PROTECTED DOCUMENT**

© ISO 2017, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Ch. de Blandonnet 8 • CP 401  
CH-1214 Vernier, Geneva, Switzerland  
Tel. +41 22 749 01 11  
Fax +41 22 749 09 47  
copyright@iso.org  
www.iso.org

# Contents

Page

<b>Foreword</b> .....	<b>iv</b>
<b>Introduction</b> .....	<b>v</b>
<b>1 Scope</b> .....	<b>1</b>
<b>2 Normative references</b> .....	<b>1</b>
<b>3 Terms and definitions</b> .....	<b>1</b>
<b>4 Software validation discussion</b> .....	<b>1</b>
4.1 Definition.....	1
4.2 Confidence-building activities: Tools in the toolbox.....	1
4.3 Critical thinking.....	2
<b>5 Software validation and critical thinking</b> .....	<b>2</b>
5.1 Overview.....	2
5.2 Determine if the software is in scope.....	6
5.2.1 Document a high-level definition of the process and use of the software.....	6
5.2.2 Regulatory use assessment.....	6
5.2.3 Processes and software extraneous to medical device regulatory requirements...	6
5.3 Development phase.....	7
5.3.1 Validation planning.....	7
5.3.2 Define.....	7
5.3.3 Implement, test and deploy.....	11
5.4 Maintain phase.....	13
5.4.1 Entering the maintenance phase.....	13
5.4.2 Planning for maintenance.....	14
5.4.3 Types of maintenance within the maintain phase.....	15
5.4.4 Process changes: Change to risk control measures.....	15
5.4.5 Emergency change.....	15
5.4.6 Maintaining for intended use.....	16
5.5 Retirement phase.....	16
<b>6 Documentation</b> .....	<b>16</b>
<b>7 Prerequisite processes</b> .....	<b>17</b>
<b>Annex A (informative) Toolbox</b> .....	<b>18</b>
<b>Annex B (informative) Risk management and risk-based approach</b> .....	<b>24</b>
<b>Annex C (informative) Examples</b> .....	<b>28</b>
<b>Bibliography</b> .....	<b>84</b>

## Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see [www.iso.org/directives](http://www.iso.org/directives)).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO should not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see [www.iso.org/patents](http://www.iso.org/patents)).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: [www.iso.org/iso/foreword.html](http://www.iso.org/iso/foreword.html).

This document was prepared by Technical Committee ISO/TC 210, *Quality management and corresponding general aspects for medical devices*, in collaboration with Technical Committee IEC/TC 62, *Electrical equipment in medical practice*, Subcommittee SC 62A, *Common aspects of electrical equipment used in medical practice*, in accordance with ISO/IEC mode of cooperation 4.

A list of all parts in the ISO 80002 series can be found on the ISO website.

## Introduction

This document has been developed to assist readers in determining appropriate activities for the validation of process software used in medical device quality systems using a risk-based approach that applies critical thinking.

This includes software used in the quality management system, software used in production and service provision, and software used for the monitoring and measurement of requirements, as required by ISO 13485:2016: 4.1.6, 7.5.6 and 7.6.

This document is the result of an effort to bring together experience from medical device industry personnel who deal with performing this type of software validation and who are tasked with establishing auditable documentation. The document has been developed with certain questions and problems in mind that we all go through when faced with validating process software used in medical device quality systems such as the following: What has to be done? How much is enough? How is risk analysis involved? After much discussion, it has been concluded that in every case, a set of activities (i.e. the tools from a toolbox) was identified to provide a level of confidence in the ability of the software to perform according to its intended use. However, the list of activities varied depending on factors including, among others, the complexity of the software, the risk of harm involved and the pedigree (e.g. quality, stability) of vendor-supplied software.

The intention of this document is to help stakeholders, including manufacturers, auditors and regulators, to understand and apply the requirement for validation of software included in ISO 13485:2016, 4.1.6, 7.5.6 and 7.6.



# Medical device software —

## Part 2:

# Validation of software for medical device quality systems

### 1 Scope

This document applies to any software used in device design, testing, component acceptance, manufacturing, labelling, packaging, distribution and complaint handling or to automate any other aspect of a medical device quality system as described in ISO 13485.

This document applies to

- software used in the quality management system,
- software used in production and service provision, and
- software used for the monitoring and measurement of requirements.

It does not apply to

- software used as a component, part or accessory of a medical device, or
- software that is itself a medical device.

### 2 Normative references

There are no normative references in this document.

### 3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 9000 and ISO 13485 apply.

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>

### 4 Software validation discussion

#### 4.1 Definition

The term “software validation” has been interpreted both broadly and narrowly, from just testing to extensive activities including testing. This document uses the term software validation to denote all of the activities that establish a level of confidence that the software is appropriate for its intended use and that it is trustworthy and reliable. The chosen activities, whatever they might be, should ensure that the software meets its requirements and intended purpose.

#### 4.2 Confidence-building activities: Tools in the toolbox

The tools in the toolbox (see [Table A.1](#) to [Table A.5](#)) include activities completed during the life cycle of software that reduce risk and build confidence.

### 4.3 Critical thinking

This document promotes the use of critical thinking to determine which activities should be performed to adequately validate specific software. Critical thinking is a process of analysing and evaluating various aspects of software, as well as the environment in which it will be used, to identify the most meaningful set of confidence-building activities to be applied during validation. Critical thinking avoids an approach that applies a one-size-fits-all validation solution without thoroughly evaluating the solution to determine if it indeed results in the desired outcome. Critical thinking recognizes that validation solutions can vary greatly from software to software and also allows for different validation solutions to be applied to the same software in a similar situation. Critical thinking challenges proposed validation solutions, to ensure that they meet the intent of the quality management system requirements, and considers all key stakeholders and their needs. Critical thinking is also used to re-evaluate the validation solution when characteristics of the software change, when the software's intended use changes or when new information becomes available.

Critical thinking results in a validation solution that establishes compliance for a manufacturer, ensures that the software is safe for use, results in documented evidence that is deemed appropriate and adequate by reviewers, and results in a scenario in which individuals performing the validation work feels that the effort adds value and represents the most efficient way to reach the desired results.

[Annex C](#) presents example studies demonstrating how critical thinking can be applied to software validation of software used in medical device quality systems in a variety of situations, including different complexities, pedigrees and risk levels.

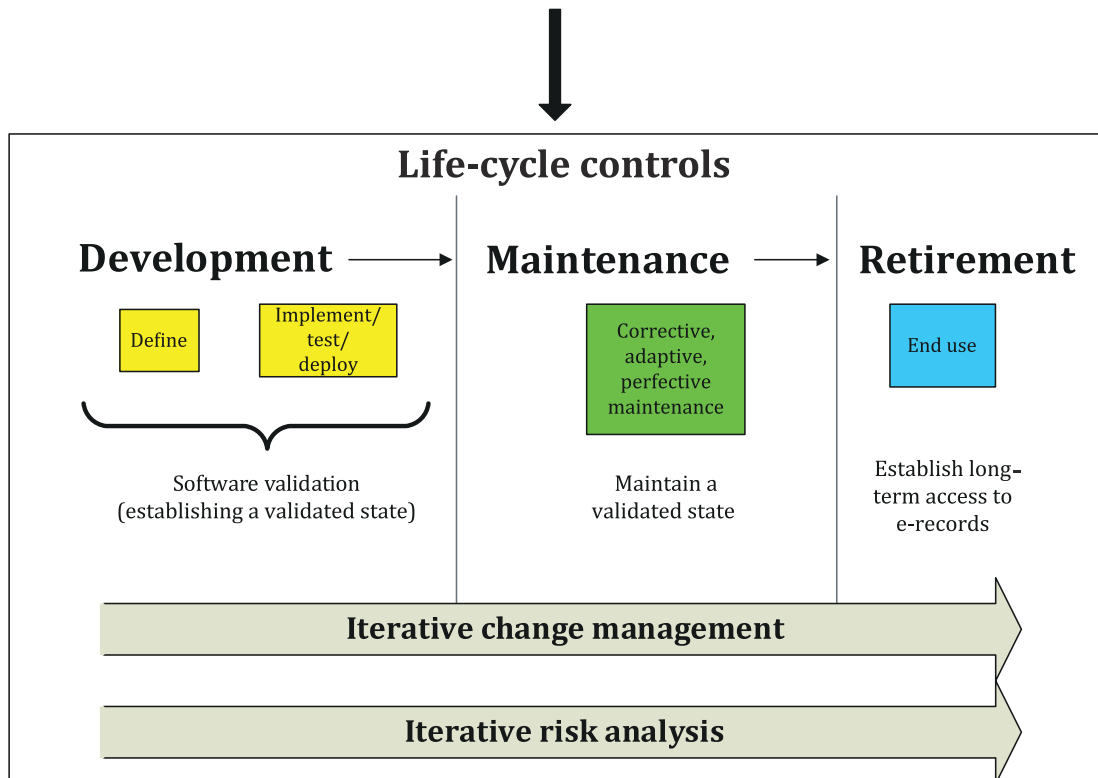
## 5 Software validation and critical thinking

### 5.1 Overview

Throughout the life cycle of software for medical device quality systems, appropriate controls need to be in place to ensure that the software performs as intended. Incorporation of critical thinking and application of selected confidence-building activities result in establishing and maintaining a validated state of the software. [Figure 1](#) depicts a conceptual view of typical activities and controls that are part of the life cycle from the moment the decision is made to automate a process until the software is retired or is no longer used for medical device quality systems. Although [Figure 1](#) depicts a sequential model, in reality, the process is of an iterative nature as elements are defined, risks are identified and critical thinking is applied.

When developing software for use in the medical device quality system, a fundamental confidence-building activity to be selected from the toolbox is the choice of software development life-cycle model. The model chosen should include critical thinking activities that enable the selection of other appropriate tools during various life-cycle activities. The results of the analyses and evaluations used drive the selection of the most meaningful set of confidence-building activities to ensure that the software performs as intended. This document does not mean to imply or prescribe the use of any particular software development model. For simplicity, however, the remainder of this document explains the concepts of critical thinking within the context of a waterfall development model using generic names for the phases. Other software development models (e.g. iterative, spiral) can certainly be used as long as critical thinking and the application of appropriate tools are incorporated into the model.

## Software for medical device quality systems



**Figure 1 — Life-cycle controls**

When considering using software in a process, one should identify whether the proposed software is used as part of a medical device quality system process through an investigation of its intended use. If so, then the software should be validated for its intended use. Although this document describes an approach to validating software for medical device quality systems, the same approach is also good practice for software to evaluate whether it fulfils defined requirements. The most critical part of software validation is developing/purchasing the right software tool to be able to support processes as intended by the manufacturer. This implies that requirements should be determined accurately to evaluate whether the developed/purchased software is suitable to fulfil the requirements of the intended use. Technical requirements suitable for verification, as well as process requirements suitable for validation, are equally important. When considering using software in a process, the software can interact or can have interfaces with other software.

During the development phase of the life cycle, risk management and validation planning tasks are performed to gather information and drive decisions in the following four areas:

- level of effort applied and scrutiny of documentation and deliverables;
- extent of content in the documentation and deliverables;
- selection of tools from the toolbox and methods for applying the tools;
- level of effort in applying the tools.

The primary drivers for decisions in the four areas are process risk and software risk. However, other drivers can influence decisions, including the complexity of the software and process, the type of software and the software pedigree.

The validation planning process consists of two distinct elements. The first validation planning element involves determining the level of rigor in the documentation and the scrutiny to be applied to the review of the resulting deliverables. The decisions in this element are primarily driven by the results

of the process risk analysis. The second validation planning element drives the selection of tools from the toolbox to implement, test and deploy the software. The choice of tools is driven primarily by the software risk analysis. Such planning steps result from different types of risk analyses and are depicted as separate activities in this document. However, many times the steps are combined into one activity, which includes the different aspects of risk analysis and the resultant choices for proceeding with validation.

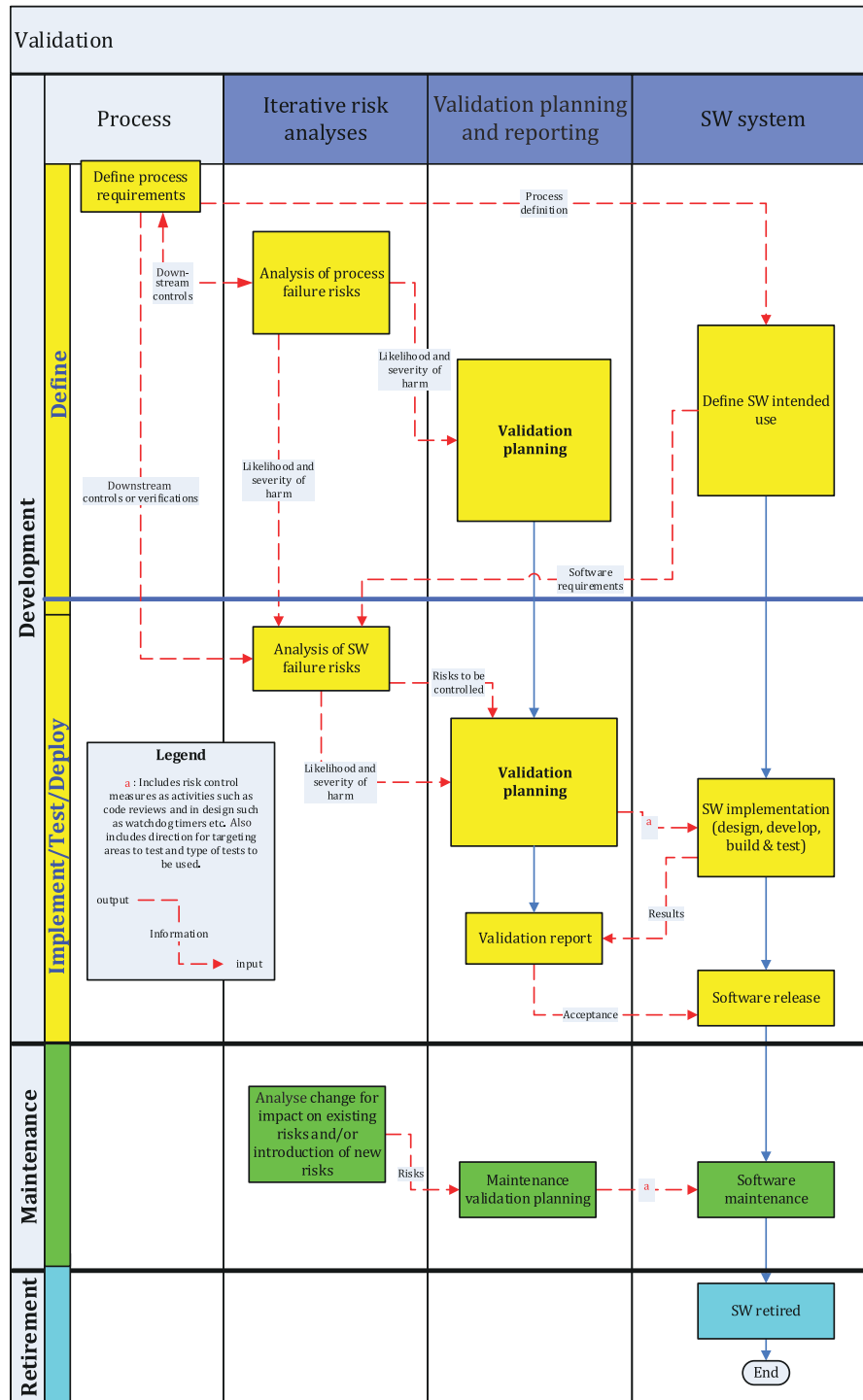
During the development phase of the life cycle, risk management and validation planning tasks are used to define the appropriate level of effort to be applied to the software and to determine what confidence-building tools to apply. This type of approach results in the completion of appropriate value-added activities and verification tasks, which are the basis for establishing a validated state. Once these activities and tasks are executed, the tools and their associated results are cited in a validation report as support for the conclusion that the software is validated.

Once deployed, the software moves into the maintenance phase of the software life cycle. During this period, the software is monitored, enhanced and updated as dictated by the business needs or regulatory requirement changes. Change control activities use the same concepts as the initial approach that was applied during the development phase of the life cycle. Changes, however, are now assessed as to their effect on the intended use, on the risk of failure, on the risk control measures that were applied during the initial development and on any functionality of the software itself.

The retirement phase is the act of removing software from use either by removal of the process or by replacement of the software being used for the process.

The activities shown in [Figure 1](#) reflect the primary software life-cycle control activities. Other work streams include project management, process development, vendor management (if applicable), and possibly others, depending on the software being implemented.

[Figure 2](#) depicts software life-cycle control activities and critical thinking within the context of other work stream activities. The critical thinking activities appear in the iterative risk analysis and validation work streams. It is important to have clear and formal definitions of these work streams within the organization's business model to ensure that a program properly manages the software from both business and regulatory perspectives.



NOTE When the term "develop" or "development" is used, it is about the development of a validated state of the software.

Figure 2 — Life-cycle controls work stream

The various colours depicted in Figure 2 correspond to the life-cycle portion that is shown in the overall approach flow chart in Figure 1. The red dashed lines indicate information that is outputted from one activity and that provides input to or helps drive decisions in another activity. The diagram demonstrates how the ordering of the activities is driven by the need to have input information before completing the activities that require the input. It is important to note that all the activities are completed irrespective of the size or complexity of the software being implemented. However, for