

Software and systems engineering — Requirements for designers and developers of user documentation

ICS 35.080

National foreword

This British Standard is the UK implementation of ISO/IEC 26514:2008. It supersedes BS 7137:1989 and BS ISO/IEC 18019:2004 and BS ISO/IEC 6592:2000 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/15, Software and systems engineering.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 July 2008

© BSI 2008

ISBN 978 0 580 58205 9

Amendments/corrigenda issued since publication

Date	Comments

INTERNATIONAL STANDARD

BS ISO/IEC 26514:2008

ISO/IEC 26514

First edition
2008-06-15

Systems and software engineering — Requirements for designers and developers of user documentation

*Ingénierie du logiciel et des systèmes — Exigences pour les
concepteurs et les développeurs de la documentation de l'utilisateur*



Reference number
ISO/IEC 26514:2008(E)

AVA Documents
Document Sharing Hub
© ISO/IEC 2008

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2008

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents		Page
Foreword.....		viii
Introduction		ix
1	Scope	1
2	Conformance	3
2.1	Application of conformance	3
2.2	Conformance situations.....	3
3	Normative references	4
4	Terms and definitions.....	4
5	User documentation process within the systems/software life cycle.....	10
6	Project requirements, objectives, and constraints	13
6.1	Project objectives	13
6.2	User documentation requirements and constraints	14
6.3	Project goals and constraints	15
6.3.1	Project infrastructure and tools	16
6.3.2	Schedule constraints.....	16
6.4	Users and usability objectives	17
6.5	Interviewing technical contacts and other experts	19
6.6	Project planning	19
6.6.1	Quality management.....	20
6.6.2	Version control and change control	21
6.6.3	Availability of resources	21
6.6.4	Schedules	22
6.6.5	Cost estimating	23
6.6.6	Planning for localization and customization	24
6.7	Documentation proposal	24
7	Analysis and design	25
7.1	Audience and task analysis	25
7.1.1	Audience analysis.....	25
7.1.2	Audience profiles.....	27
7.1.3	Task analysis.....	28
7.2	User documentation design	30
7.2.1	Designing for use of content	31
7.2.2	Designing formats	31
8	Development and review.....	32
8.1	Prototypes and drafts.....	32
8.1.1	CM during development.....	33
8.1.2	Development of translated and localized documentation.....	33
8.2	Evaluation of documentation	34
8.2.1	Other roles in evaluation of documentation quality.....	35
8.2.2	Documentation review procedures.....	36
8.3	Documentation testing	38
8.3.1	Types of documentation tests	38
8.3.2	Usability tests.....	39
9	Production	40
9.1	Final assembly and review	40
9.2	Approval	40
9.3	CM.....	41

9.4	Updating and maintenance	41
10	Structure of documentation	41
10.1	Overall structure of documentation	42
10.1.1	Structure of instructional mode documentation.....	43
10.1.2	Structure of reference mode documentation	43
10.2	Structure of documentation according to audience needs	43
10.3	Size of topics in onscreen documentation.....	45
10.4	User documentation components	46
10.5	Placement of user documentation components.....	47
10.5.1	Initial components	47
10.5.2	Placement of critical information	47
11	Information content of user documentation	47
11.1	Completeness of information	48
11.2	Accuracy of information.....	48
11.3	Content of identification data	48
11.4	Information for use of the documentation.....	49
11.5	Concept of operations	50
11.6	Information for general use of the software.....	51
11.7	Information for procedures and tutorials	52
11.7.1	Preliminary information for procedures	52
11.7.2	Procedural steps	52
11.7.3	Completion information for procedures	53
11.7.4	Tutorials	54
11.8	Information on software commands	54
11.9	Explanations of data entry fields.....	55
11.10	Content of error messages and problem resolution	55
11.11	Content of warnings and cautions	56
11.12	Information on terminology	57
11.13	Information on related information sources	57
11.14	User-supplied content	58
12	Presentation format of documentation	59
12.1	General	59
12.2	Use of printed or on-screen formats.....	60
12.3	Selection of appropriate media and format.....	61
12.3.1	Comparison of media	61
12.3.2	Relationship of information displays to the application's displays	63
12.4	Context-sensitive information	63
12.5	Accessible documentation.....	64
12.5.1	Provide understandable documentation	64
12.5.2	Provide user documentation in accessible electronic form.....	65
12.5.3	Provide text alternatives in on-screen documentation	65
12.5.4	Write instructions without unnecessary device references	65
12.5.5	Provide documentation on accessibility features	65
12.6	Consistency of formats	65
12.7	Consistency of terminology.....	66
12.8	Layout of screens and pages.....	67
12.8.1	Grids	67
12.8.2	Non-scrolling areas.....	68
12.8.3	Arrangement of windows	68
12.8.4	Formats for information area (text)	69
12.8.5	Formats for headings	69
12.8.6	Blank space and borders	69
12.8.7	Vertical spacing.....	70
12.9	Legibility.....	70
12.9.1	Typefaces and text size	71
12.9.2	Highlighting text.....	71
12.9.3	Lines of text	72
12.10	Formats for lists	72

12.11	Formats for representing user interface elements.....	73
12.11.1	Representing control and command input	73
12.11.2	Representing special keyboard keys.....	73
12.12	Use of color	74
12.13	Navigational features.....	75
12.13.1	Using formats to indicate position within a topic	75
12.13.2	Finding the same information again	76
12.13.3	Viewing topics in sequence	76
12.13.4	Formats for active areas	76
12.13.5	Linking information	77
12.14	Documentation formats for finding information.....	77
12.14.1	Table of contents	77
12.14.2	Menus.....	78
12.14.3	List of illustrations	79
12.14.4	Index.....	79
12.14.5	Search capability	80
12.15	Formats for warnings, cautions, and notes	80
12.16	Format for instructions	81
12.17	Formats for user-supplied annotations.....	81
12.18	Formats for illustrations	81
12.18.1	When to use an illustration.....	81
12.18.2	Level of detail in illustrations	82
12.18.3	Identification of illustrations.....	82
12.18.4	Consistent presentation of illustrations.....	82
12.18.5	Placement of illustrations	83
12.18.6	Illustrations of screen displays	83
12.18.7	Illustrations of printed output.....	84
12.18.8	Tables.....	85
12.19	Icons and signposts	85
12.19.1	When to use icons and signposts.....	85
12.19.2	Design of icons and signposts.....	85
12.19.3	Displaying the names of icons	86
12.20	Documentation packaging.....	87
Annex A	(informative) User documentation style guide content	88
A.1	Writing style	88
A.2	Language	88
A.3	Spelling	88
A.4	Grammar and usage	88
Annex B	(informative) Writing style and techniques for user documentation.....	89
B.1	General.....	89
B.2	Style for sentences	90
B.3	Style for paragraphs	94
B.4	Style for quick-reference information.....	95
B.5	Style for installation instructions	95
B.6	Style for tutorials and task instructions	96
B.7	Style for describing user interface elements	96
B.8	Style for descriptions and explanations.....	96
B.9	Style for on-screen information	97
B.10	Style for lists	97
Annex C	(informative) User documentation style for translation and localization.....	98
C.1	General.....	98
C.2	Terminology	98
C.3	Style for translation	99
C.4	Cultural factors	100
Annex D	(informative) Design, development, and production of printed information	102
D.1	Introduction	102
D.2	Design	102
D.3	Production phase.....	110

Annex E (informative) Checklists for user documentation	114
E.1 Checklist for printed manuals.....	114
E.2 Checklist for online help	117
Annex F (informative) Requirements clauses and checklist for the documentation process	121
Annex G (informative) Requirements clauses and checklist for documentation products.....	127
Bibliography	141

List of Figures

Figure 1 — Sample process of defining the usability goals for an electronic mail system	18
Figure 2 — Sample contents list for a documentation proposal	25
Figure 3 — Sample list of audiences for part of an order fulfillment system.....	26
Figure 4 — Sample of an audience hierarchy.....	26
Figure 5 — Sample audience profile.....	28
Figure 6 — Sample task list for an electronic mail system	28
Figure 7 — Sample task hierarchy.....	29
Figure 8 — Using audience information needs to determine document content.....	44
Figure 9 — Using information type and usage to determine delivery method	45
Figure 10 — Sample presentation of an example.....	50
Figure 11 — Sample overview for a software product module	51
Figure 12 — Sample function description for a spreadsheet function	55
Figure 13 — Sample definition of a term	57
Figure 14 — Sample links to related information	58
Figure 15 — Sample grid for a help system navigator and a topic window	68
Figure 16 — Example of a contents list	78
Figure 17 — Sample text menu	79
Figure 18 — Use of two scales for screen displays	84
Figure D.1 — Example of an A5 page grid	107
Figure D.2 — Example of an A5 page	108

List of Tables

Table 1 — Audience mapping matrix.....	30
Table 2 — Components of documentation	46

Table 3 — Example of procedures with the elements marked	53
Table 4 — Sample Information Profile for one task and one audience	61
Table 5 — Advantages and disadvantages of various media	62
Table 6 — Examples of access methods	64
Table B.1 — Example of conditions presented as a table	91
Table D.1 — Methods of producing multiple copies	103

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 26514 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

Introduction

Anyone who uses application software needs accurate information about how the software will help the user accomplish a task. The documentation may be the first tangible item that the user sees and therefore influences the user's first impressions of the software product. If the information is supplied in a convenient form and is easy to find and understand, the user can quickly become proficient at using the product. Hence, well-designed documentation not only assists the user and helps to reduce the cost of training and support, but also enhances the reputation of the product, its producer, and its suppliers.

Although software developers aim to design user interfaces that behave so intuitively that very little separate documentation is needed, this is rarely possible. Today's software offers increasingly robust functionality, not only within applications, but also across applications that intelligently exchange information with one another. Further, most software designs include underlying rules and calculations, or algorithms, that affect the results a user can obtain when using the software. Such underlying programming mechanics are discernable by users, but only through laborious testing. For these reasons and more, user documentation remains an essential component of usable software products.

Documentation is often regarded as something done after the software has been implemented. However, for high-quality software documentation, its development should be regarded as an integral part of the software life cycle process. If done properly, documentation or information management is a big enough job to require process planning in its own right.

This International Standard was developed to assist users of ISO/IEC 15288:2008, *Systems and software engineering — System life cycle processes*, or ISO/IEC 12207:2008, *Systems and software engineering — Software life cycle processes*, to design and develop documentation as part of the software life cycle processes. It defines the documentation process from the documentation developer's standpoint.

NOTE Other International Standards in the ISO/IEC 265NN family are in preparation or planned to address the documentation and information management processes from the viewpoints of managers, assessors and testers, and acquirers and suppliers.

In addition to defining a standard process, this International Standard also covers the documentation product. This International Standard specifies the structure, content, and format for documentation, and also provides informative guidance for user documentation style.

Earlier standards tended to view the results of the documentation process as a single book or multivolume set: a one-time deliverable. Increasingly, documentation designers recognize that most user documentation is now produced from managed re-use of previously developed information (single-source documentation), adapted for new software versions or presentation in various on-screen and printed media. While this International Standard does not describe how to set up a content management system (CMS), it is applicable for documentation organizations practicing single-source documentation.

This International Standard is independent of the software tools that may be used to produce documentation, and applies to both printed documentation and on-screen documentation. Much of its guidance is applicable to user documentation for systems including hardware as well as software user documentation.

This International Standard conforms to ISO/IEC 12207:2008 as an implementation of subclause 7.2.1, Software Documentation Management Process, for software user documentation. This International Standard may be used as a conformance or a guidance document for documentation products, projects, and organizations claiming conformance to ISO/IEC 15288:2008 or to ISO/IEC 12207:2008.

The primary sources for this International Standard are previous standards IEEE Std 1063-2001, *IEEE standard for software user documentation*, and ISO/IEC 18019:2004, *Software and system engineering — Guidelines for the design and preparation of user documentation for application software*.

Systems and software engineering — Requirements for designers and developers of user documentation

1 Scope

This clause presents the scope, purpose, organization, and candidate uses of this International Standard.

This International Standard supports the interest of software users in consistent, complete, accurate, and usable documentation. It includes both approaches to standardization: a) process standards, which specify the way in which documentation products are to be developed; and b) documentation product standards, which specify the characteristics and functional requirements of the documentation.

The first part of this International Standard covers the user documentation process for designers and developers of documentation. It describes how to establish what information users need, how to determine the way in which that information should be presented to the users, and how to prepare the information and make it available. It is not limited to the design and development phase of the life cycle, but includes activities throughout the information management and documentation processes.

The second part of this International Standard provides minimum requirements for the structure, information content, and format of user documentation, including both printed and on-screen documents used in the work environment by users of systems containing software. It applies to printed user manuals, online help, tutorials, and user reference documentation.

This International Standard neither encourages nor discourages the use of either printed or electronic (on-screen) media for documentation, or of particular documentation development or management tools or methodologies.

This International Standard may be helpful for developing the following types of documentation, although it does not cover all aspects of them:

- documentation of products other than software;
- multimedia systems using animation, video, and sound;
- computer-based training (CBT) packages and specialized course materials intended primarily for use in formal training programs;
- documentation produced for installers, computer operators, or system administrators who are not end users;
- maintenance documentation describing the internal operation of systems software;
- documentation incorporated into the user interface itself.

This International Standard is applicable to documentation designers and developers, including a variety of specialists:

- information designers and architects who plan the structure and format of documentation products in a documentation set;
- usability specialists and business analysts who identify the tasks that the intended users will perform with the software;
- those who develop and edit the written content for user documentation;
- graphic designers with expertise in electronic media;
- user interface designers and ergonomics experts working together to design the presentation of the documentation on the screen.

This International Standard may also be consulted by those with other roles and interests in the documentation process:

- managers of the software development process or the documentation process;
- acquirers of documentation prepared by suppliers;
- usability testers, documentation reviewers, subject-matter experts;
- developers of tools for creating on-screen documentation;
- human-factors experts who identify principles for making documentation more accessible and easily used.

This International Standard is intended for use in all types of organizations, whether or not a dedicated documentation department is present, and may be used as a basis for local standards and procedures. Readers are assumed to have experience or knowledge of software development or documentation development processes.

Users of this International Standard should adopt a style manual for use within their own organizations to complement the guidance provided in the annexes to this International Standard, or adopt an industry-recognized style guide. Annex A provides guidance for the content of a style guide, and Annexes B and C provide guidance on style.

The order of clauses in this International Standard does not imply that the documentation should be developed in this order or presented to the user in this order.

In each clause, the requirements are media-independent, as far as possible. Requirements specific to either print or electronic media are identified as such, particularly in Clause 12. Annex D provides guidance for the design of printed documentation.

The checklists in Annex E may be used at each phase of the documentation process to check that the appropriate steps have been carried out and that the finished documentation satisfies quality criteria.

The checklists in Annexes F and G may be used to track conformance with the requirements of this International Standard for documentation processes and products.

The bibliography lists works that provide guidance on the processes of managing, preparing, and testing user documentation.

2 Conformance

This International Standard may be used as a conformance or a guidance document for projects and organizations claiming conformance to ISO/IEC 15288:2008 or ISO/IEC 12207:2008.

2.1 Application of conformance

Whether the organization or project has tailored the selected software life cycle processes or adopted them in full, the organization or project may claim conformance to this International Standard for its documentation process, for the documentation, or for both.

This International Standard is meant to be tailored so that only necessary and cost-effective requirements are applied to documentation. Tailoring may take the form of specifying approaches to conform to its mandatory requirements, or altering its non-mandatory recommendations and approaches to reflect the particular software and documentation product more explicitly. Tailoring decisions made by the acquirer should be specified in the contract.

Throughout this International Standard, “shall” is used to express a provision that is binding, “should” to express a recommendation among other possibilities, and “may” to indicate a course of action permissible within the limits of this International Standard. When using this International Standard as a guide, replace the term “shall” with “should”. Use of the nomenclature of this International Standard for the parts of user documentation (that is, chapters, topics, pages, screens, windows) is not required to claim conformance.

NOTE All “shall” statements are listed in Annex F and Annex G.

2.2 Conformance situations

Conformance of user documentation may be interpreted differently for various situations. The relevant situation shall be identified in the claim of conformance:

- 1) When conformance is claimed for an organization, the organization shall make public a document declaring its tailoring of the life cycle process.

NOTE 1 One possible way for an organization to deal with clauses that cite “the documentation plan” is to specify that they shall be interpreted in the project plans for any particular documentation project.

- 2) When conformance is claimed for a project, the project plans or the contract shall document the tailoring of the documentation requirements.

NOTE 2 A project’s claim of conformance is typically specified with respect to the organization’s claim of conformance.

- 3) When conformance is claimed for a multi-supplier program, it may be the case that no individual project may claim conformance because no single contract calls for all the required activities. Nevertheless, the program, as a whole, may claim conformance if each of the required activities is produced by an identified party. The program plans shall document the tailoring of the required tasks, and their assignment to the various parties, as well as the interpretation of clauses of this International Standard that reference “the contract”.
- 4) When conformance is claimed for documentation products, the organization or project should specify whether conformance applies to a single document, a documentation set, or all user documentation produced through the organization’s content management processes.

This International Standard may be included or referenced in contracts or similar agreements when the parties (called the acquirer and the producer or supplier) agree that the supplier will deliver documentation in accordance with this International Standard. It may also be adopted as an in-house standard by a project or organization that decides to produce documentation in accordance with this International Standard.

3 Normative references

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

ISO/IEC 12207:2008, *Systems and software engineering — Software life cycle processes*

ISO/IEC 15288:2008, *Systems and software engineering — System life cycle processes*

IEEE Std 100-2000, *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition

4 Terms and definitions

For the purposes of this document, the terms and definitions given in IEEE Std 100-2000 and the following apply.

NOTE Throughout this International Standard, the term “documentation” refers to software user documentation. Use of the terminology in this International Standard is for ease of reference and is not mandatory for conformance with this International Standard.

4.1 accessibility

usability of a software or documentation product, service, environment, or facility by people with the widest range of capabilities

NOTE 1 Adapted from ISO 9241-171.

NOTE 2 Although “accessibility” typically addresses users who have disabilities, the concept is not limited to disability issues.

4.2 action

element of a step that a user performs during a procedure

4.3 active area

⟨on-screen documentation⟩ area that responds to user input

NOTE A hot-spot on a graphic and a link in text are examples of active areas.

4.4 analysis

investigation and collection phase of development that aims to specify types of users and their informational needs

4.5 application software

software designed to help users perform particular tasks or handle particular types of problems, as distinct from software that controls the computer itself

cf. **software**

4.6 audience

category of users sharing the same or similar characteristics and needs (for example, reason for using the documentation, tasks, education level, abilities, training, experience)

NOTE There may be different audiences for documentation (for example, management, data entry, maintenance) that determine the content, structure, and use of the intended documentation.

4.7

caution

advisory information in documentation that states that performing some action may lead to consequences that are unwanted or undefined, such as loss of data or an equipment problem

cf. **warning**

4.8

change control procedure

actions taken to identify, document, review, and authorize changes to a software or documentation product that is being developed

NOTE The procedures ensure that the validity of changes is confirmed, that the effects on other items are examined, and that those people concerned with the development are notified of the changes.

4.9

configuration management

CM

technical and organizational activities comprising configuration identification, control, status accounting, and auditing

NOTE See ISO 10007:2003, *Quality management systems — Guidelines for configuration management*.

4.10

context-sensitive help

type of on-screen documentation in which the information that is displayed depends upon the user's view of the software

cf. **embedded documentation, printed documentation**

4.11

critical information

information describing the safe use of the software, the security of the information created with the software, or the protection of the sensitive personal information created by or stored with the software

4.12

customization

adaptation of a software or documentation product to the needs of a particular audience

4.13

design

stage of documentation development that is concerned with determining what documentation will be provided in a product and what the nature of the documentation will be

4.14

development

activity of preparing documentation after it has been designed

4.15

document

separately identified piece of documentation which could be part of a documentation set

4.16

documentation

information that explains how to use a software product

NOTE 1 It can be provided as separate documentation or as embedded documentation or both.

NOTE 2 In this International Standard, the term “documentation” is synonymous with the terms “user documentation” and “software user documentation”. Other forms of documentation (for example, “system documentation”) are clearly identified as such.

NOTE 3 Printed manuals, on-screen information and stand-alone online help are examples of documentation.

4.17

document set

collection of documentation that has been segmented into separately identified volumes or files for ease of distribution or use

4.18

embedded documentation

documentation that is accessed as an integral part of software

cf. **separate documentation**

NOTE Pop-up help and help text on a screen are examples of embedded documentation.

4.19

entry field

area on a screen or in a window in which a user enters data

4.20

escrow

source code and documentation that is kept in the custody of a third party until specified contractual conditions have been fulfilled

4.21

function

part of an application that provides facilities for users to carry out their tasks

4.22

icon

graphic displayed on the screen that represents a function of the computer system

NOTE Adapted from ISO/IEC 11581-1:2000, 4.7.

4.23

illustration

graphic set apart from the main body of text and normally cited within the main text

NOTE In this International Standard, the term “illustration” is used as the generic term for tables, figures, exhibits, screen captures, flow charts, diagrams, drawings, icons, and other types of graphics.

4.24

implementation

phase of development during which user documentation is created according to the design, tested, and revised

4.25

instructional mode

usage mode that is intended to teach the use of software in performing tasks

4.26

internationalization

process of developing information so that it is suitable for an international audience

cf. **localization**

4.27

link

⟨on-screen documentation⟩ active area that displays either a new topic or a different part of the current topic

4.28

localization

creation of a national or specific regional version of a product

cf. **internationalization**

NOTE Localization may be carried out separately from the translation process.

4.29

menu

⟨on-screen documentation⟩ list of topics from which the user may choose

4.30

navigation

act of accessing documentation and viewing different topics

4.31

on-screen documentation

documentation that is intended to be read on the screen by the user while using the software

cf. **printed documentation, embedded documentation**

NOTE Pop-up help and help text on a screen are examples of on-screen documentation.

4.32

picture

illustration that shows the actual appearance of physical objects

NOTE Photographs and drawings are examples of pictures.

4.33

printed documentation

documentation that is either provided in printed form, or provided in electronic form for the customer or user to print

4.34

procedure

ordered series of steps that specify how to perform a task

4.35

process

set of interrelated or interacting activities which transforms inputs into outputs

[ISO 9000:2005]

4.36

product

complete set of software and documentation

4.37

product authority

person or persons with overall responsibility for the capabilities and quality of a product

4.38

project

set of activities for developing a new product or enhancing an existing product

4.39

project manager

person with overall responsibility for the management and running of a project

4.40

quality management

coordinated activities to direct and control an organization with regard to quality

[ISO 9000:2005]

4.41

real-world object

entity that exists in a three-dimensional form and, by association, implies similar properties or behavior to software functions

NOTE Printer, filing cabinet, file folder and sheet of paper are examples of real-world objects.

4.42

reference mode

usage mode that is intended to provide quick access to specific information for software users who are generally familiar with the software functions

4.43

secondary window

window that contains information that depends on information in another window (the primary window)

NOTE The information in the secondary window supplements the information in the primary window.

4.44

separate documentation

documentation that may be used independently of the software

cf. **embedded documentation**

NOTE Printed manuals and freestanding hypertext systems are examples of separate documentation.

4.45

signpost

text, symbol, or small graphic that helps the user identify where particular types of information are located or where the information in the current display fits into the whole document

NOTE Information of different types may be indicated by symbols or graphics of different types.

4.46

software

program or set of programs used to run a computer

cf. **application software**

NOTE For the purposes of this International Standard, the term “software” does not include on-screen documentation.

4.47

step

one element (numbered list item) in a procedure that tells a user to perform an action (or actions)

NOTE Responses by the software are not considered to be steps.

4.48

style

set of language-specific editorial conventions covering grammar, terminology, punctuation, capitalization, and word choice of documentation

4.49

technical contact

person responsible for providing a documentation developer with technical information about a software product or for checking the technical accuracy of drafts of documentation; also called subject-matter expert

4.50

topic

small part of a document that deals with a single subject

NOTE 1 In printed documentation, a topic is equivalent to a section (heading; subheading) and its content. In on-screen documentation, a topic consists of a title (heading) and information about a subject (typically, a task or a concept or reference information).

NOTE 2 For on-screen documentation, the system may present a topic without user intervention.

NOTE 3 Instructions on how to print the current document are an example of a topic.

4.51

tutorial

instructional mode documentation in which the user exercises software functions using sample data that is supplied with the software or documentation

4.52

usability

extent to which a software or documentation product may be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use

NOTE Adapted from ISO 9241-11:1998, 3.1.

4.53

usage mode

primary manner in which the documentation developer expects the document to be used

NOTE This International Standard recognizes two usage modes: instructional and reference.

4.54

user

person who performs one or more tasks with software; a member of a specific audience

4.55

user documentation

information to describe, explain, or instruct how to use software

cf. **documentation**

NOTE Printed user manuals, embedded on-screen information, and help are examples of user documentation.

4.56

user interface

ensemble of software and hardware that allows a user to interact with a computer system

4.57

warning

advisory information in documentation that states that performing some action may lead to serious or dangerous consequences

cf. **caution**

4.58

window

area with visible boundaries that presents a view of a software object or through which a user conducts a dialog with a computer system

4.59

wizard

procedural form of help that guides a user through each step of a task through dialog with the user

5 User documentation process within the systems/software life cycle

This International Standard covers the phases involved in designing, specifying, and producing user documentation.

Designers and developers of user documentation work within the life cycle processes of the software product, which is defined in ISO/IEC 12207:2008 and includes the management of software documentation within the software support processes (*Clause 7.2.1*).

Thus, development of the user documentation should be part of the same processes as the software product life cycle, and ideally performed in conjunction with development of the software, so that the software and the user documentation may be tested, distributed, and maintained together. The specification of all the documentation, including on-screen documentation and printed documentation, should be a part of the development of the software product as a whole, not a separate exercise. Although accurate user documentation cannot be completed until the software product has been fully developed, the user documentation and the software product both benefit from concurrent development.

The classic documentation development process applies to the life cycle of a single new software product with a single new user manual, but it is much more likely that software and user documentation are designed and developed under more complex circumstances, such as the following:

- 1) A previously documented software product is being upgraded, offered in a new version or on different operating system platforms, or customized as part of system integration, and previous documentation must be revised.
- 2) Previous documentation must be converted to a different format or different media, or in different languages or versions, such as tutorials, online help, or advanced reference guides.
- 3) Previous documentation must be adapted or used as models for different software products acquired or supplied by an organization.

Designing and developing user documentation is greatly assisted by the presence of other documentation produced during the software life cycle, such as a documentation plan, system design document, system test plan, release records, and problem reports. Other documentation specific to the documentation process may be produced, such as style guides and organizational procedures for content management and documentation reviews.

NOTE: ISO/IEC 15289:2006, *Systems and software engineering — Content of systems and software life cycle process information products (Documentation)*, provides recommended contents for the required documents throughout the systems/software life cycle.

Thus, rather than focusing on the production of a single book or help system, user documentation designers should regard their tasks as part of the information management process, whose activities are defined in ISO/IEC 15288:2008, 6.3.6.3:

Information Management Process

a) **Plan Information Management.** This activity consists of the following tasks:

1) Define the items of information that will be managed during the system life cycle and, according to organizational policy, agreements, or legislation, maintained for a defined period beyond.

2) Designate authorities and responsibilities regarding the origination, generation, capture, archiving and disposal of items of information.

3) Define the rights, obligations and commitments regarding the retention of, transmission of and access to information items.

NOTE Due regard is paid to information and data legislation, security and privacy, e.g. ownership, agreement restrictions, rights of access, intellectual property and patents. Where restrictions or constraints apply, information is identified accordingly.

Staff having knowledge of such items of information are informed of their obligations and responsibilities.

4) Define the content, semantics, formats and medium for the representation, retention, transmission and retrieval of information.

NOTE The information may originate and may terminate in any form (e.g. verbal, textual, graphical, numerical) and may be stored, processed, replicated and transmitted using any medium (e.g. electronic, printed, magnetic, optical). Pay due regard to organization constraints, e.g. infrastructure, inter-organizational communications, distributed project working. Relevant information storage, transformation, transmission and presentation standards and conventions are used according to policy, agreements and legislation constraints.

5) Define information maintenance actions.

NOTE This includes status reviews of stored information for integrity, validity and availability and any needs for replication or transformation to an alternative medium. Consider the need to either retain infrastructure as technology changes so that archived media can be read or the need to re-record archived media using newer technology.

b) **Perform Information Management.** This activity consists of the following tasks:

1) Obtain the identified items of information.

NOTE This may include generating the information or collecting it from appropriate sources.

2) Maintain information items and their storage records according to integrity, security and privacy requirements.

NOTE Record the status of information items, e.g. version description, record of distribution, security classification. Information should be legible and stored and retained in such a way that it is readily retrievable in facilities that provide a suitable environment, and that prevent damage, deterioration and loss.

3) Retrieve and distribute information to designated parties as required by agreed schedules or defined circumstances.

NOTE Information is provided to designated parties in an appropriate form.

4) Provide official documentation as required.

NOTE Examples of official documentation are certification, accreditation, license and assessment ratings.

5) Archive designated information, in accordance with the audit, knowledge retention, and project closure purposes.

NOTE Select the media, location and protection of the information in accordance with the specified storage and retrieval periods, and with organization policy, agreements and legislation. Ensure arrangements are in place to retain necessary documentation after project closure.

6) Dispose of unwanted, invalid or unverifiable information according to organization policy, and security and privacy requirements.

The following sections include requirements and guidance for the process phases:

- process implementation:
 - setting objectives: Identifying goals, policies, and constraints, such as enterprise, project, and software product (internal) standards that will affect the document design;
 - project planning, management, and control.

NOTE: Process implementation may involve use of the acquisition and supply process when the documentation is designed and developed by a different organization than the software.

- development and review:
 - analysis and design: Drawing up the documentation designs for the project: collecting information about the software product and users, their tasks, and their needs for information, and designing documentation based on those needs;
 - development and review: Structuring the content for usability, articulating the documentation design by creating the written and graphic content, implementing the information in the chosen media, and evaluating the user documentation with the rest of the product;
- production: Packaging and making the documentation available to the users through printed or on-screen media; maintaining configuration control of master files and released versions;
- maintenance: Keeping the documentation accurate throughout the software product life cycle, including adaptations for improved usability.

For simplicity this International Standard describes the life cycle as if there were a clear starting point for developing documentation and a clear end point. There is not a single sequence of activities, however, that can be followed in all cases for all products and all types of information. For example, design and implementation activities for on-screen documentation are often closely inter-linked, as are analysis and design, and the way they link together varies between projects.

6 Project requirements, objectives, and constraints

The user documentation developer shall gather or receive information about the wider context of the whole project, to understand the requirements that affect the design for the documentation components. The purpose of the process implementation (initiation) phase is to understand the project objectives, requirements, and constraints, particularly:

- the requirements and objectives for the software product;
- the requirements and constraints for the user documentation, such as the documentation policy and standard formats and styles set by the producer of the software product;
- the constraints that may apply to the project cost, schedule, staffing, and equipment;
- the intended acquirer and end-users for the software product and the documentation, whether through contract or commercially offered to the public;
- the usability requirements.

If the stated requirements limit the design options so that users cannot be given a suitable set of documentation, the documentation designer should:

- question the requirements, explaining the reasons for the discrepancy;
- suggest alternative solutions.

The organization should keep a record of the source of each requirement, so that it may be tracked back to its origins and its validity may be reaffirmed.

6.1 Project objectives

Ideally, the project objective should be to develop a software product that satisfies user expectations and meets user needs. Documentation developers need the following information:

- Who are the users and in what contexts they will use the product?
- Are there users with special needs?
- What is the purpose of the software product? What will it do?
- Is there a previous version? If so, which features have to be changed and which have to remain the same?
- Is the product stand-alone, or is it part of a suite of products?
- When will the product be available to document? (The product schedule may help determine the types and amount of documentation.)
- Are there plans for future versions?
- On what platforms will the product run at this release? Are there plans for other platforms later?
- Is the product being developed for a specific organization or organizations? If so, will those organizations be the exclusive users?
- Will localized or customized versions of the product be required?

The project shall maintain information on the standards the software product is required to follow, such as:

- international standards (such as ISO) publications;
- national standards for the countries in which the product will be used;
- industry standards for the system on which the product will run;
- industry standards for the system on which the on-screen documentation will be viewed;
- accessibility standards and requirements;
- company, product, or operating system standards and conventions;
- restrictions on copying or modifying the software and its documentation;
- the protection of sensitive information.

6.2 User documentation requirements and constraints

The following information should be considered in the documentation plan:

- Does the organization have a formal documentation or information management policy and procedures, and if so, how will they be applied to this project?
- What metadata is required for each piece of documentation to support document management, indexing, and searchability?
- Are there organizational standard formats, templates, styles, or systems? Are they consistent with the requirements and recommendations in this International Standard? If they do not exist, they should be established consistent with this International Standard;
- Are there constraints on required content?
- Is there a need to provide background or conceptual information, such as the working procedures of the users' organizations or information about the software product itself? If so, is that information readily available?
- Are there resources for supplying the documentation developer with technical information about the product, such as technical contacts, written specifications, and the product itself (possibly development or prototype versions)?

Documentation may have to satisfy legal and regulatory requirements. The documentation designer and developer should consider the following and take legal advice if necessary:

- requirements set by national legislation;
- copyright status of the document itself;
- copyright issues for text included in the documentation from elsewhere;
- protection of sensitive data;
- acknowledgments;
- trademarks;
- escrow conditions;
- licensing;
- any presentation requirements that apply (such as use of special logotypes on the packaging and in the documentation to identify proprietary platform-compatible documentation products);
- intellectual property rights;
- warranties and guarantees.

In terms of security, the documentation developer should determine whether it is necessary to control the integrity of on-screen documentation or allow for deliberate or accidental changes to its content made by the user.

The supplier and client should also determine how the documentation will be maintained and updated.

6.3 Project goals and constraints

The business goals of a project depend on whether the software product is produced under contract or for general commercial sale. The important features of the software product and documentation's sales objectives are:

- the target number of copies of the software product to be distributed over its projected lifetime;
- the expected profile of sales over time (will demand be steady or will there be a period of peak demand?).

The documentation designer and developer should review plans for making modified versions of the software product available to users because modifications may affect the documentation design. The plans for future modifications of the application may be identified using the following questions:

- At what intervals will the software product and documentation be modified?
- What different levels of modification will there be, for example, for issuing:
 - temporary corrections;
 - interim versions;
 - major upgrades or substantially new versions.

- What are the time scales for making modifications?
- What are the cost constraints for maintenance?
- How will the documentation be updated?
- Will previous versions be maintained or removed from support?
- How will amendments be made available to existing users?

6.3.1 Project infrastructure and tools

Project constraints may include infrastructure systems and tools, schedules, cost, and staffing.

If a new software product is part of a suite of software products, tools already specified for developing, maintaining, delivering, and viewing documentation for that suite should be considered. If the customer or organization for which the software product is being developed has existing systems with which they want the new software product to be consistent, the documentation designer and developer should consider whether these existing systems dictate the use of particular documentation delivery and viewing mechanisms.

Constraints for the documentation should not be allowed to restrict the software product from handling the real problems that it should solve, and the range of hardware and operating systems for which it is being developed. The documentation designer should obtain information on the capabilities and limitations of the planned or required tools for:

- using the software, so documentation developers may find out how it operates and how users will have to use it;
- writing, laying out, and editing the text;
- writing on-screen information and compiling help files;
- drawing the illustrations;
- reviewing the documentation;
- testing the documentation;
- conducting user tests, including tests of accessibility.

6.3.2 Schedule constraints

Designers and developers need to know the following about the software product development schedule:

- When will the alpha, beta, and acceptance testing start and be completed?
- If there will be multiple releases, is there a phased or packaged approach to the releases? What features/functions will be delivered in what phase?
- Are there work processes that need to be defined to change the way business is currently being done?
- Will the software product be submitted for government certification and accreditation?
- What is the delivery date for the finished software product, and is there a worldwide release on the same day or a different release date for each region?
- How long before the delivery date does the documentation need to be ready?
- What other milestones apply to the project, such as dates for early releases?
- What are the major dependencies between different activities in the overall project?
- If the product is to be localized or customized, what are the required delivery dates for the various localized and customized versions and how long before those delivery dates does the documentation need to be ready?
- Will the documentation be translated and what is the translation schedule, including lead time, drop-off dates to translation, and delivery dates from translation?
- Will the documentation need to pass security or legal reviews prior to delivery?
- Will the documentation be printed locally but shipped internationally, and if so, what are the shipping time estimates to build into the documentation schedule to allow for physical delivery?
- If the documentation product is to be sent to a printer or published on removable media, what is the lead time required?

6.4 Users and usability objectives

Designers should be aware of usability requirements from the beginning of the project. From the start, usability should be built into the structure, content, and format of the documentation and supporting software; it cannot be added at the end of documentation development. Therefore, the usability requirements and the method of testing them shall be specified in the analysis phase when other user needs are being determined.

The usability of the documentation is an integral part of the usability of the software product. When the usability targets are set and measured for the software product, the documentation shall be treated as an integral part of the software product. The documentation developers and software developers should work together to ensure usability. Measures of the usability of the documentation, independent of the usability of the software, shall include the following:

- the time taken to learn how to do a specified task using the instructions in the documentation;
- the ability of the user to accomplish the task at hand using the documentation.

There are two aspects to ensuring that the documentation is usable:

- Applying human-centered and software ergonomics principles to the design. ISO 13407:1999, *Human-centred design processes for interactive systems*, and the ISO 9241 series [ISO 9241-1 to ISO 9241-17, *Ergonomic requirements for office work with visual display terminals (VDTs)*; ISO 9241-110:2006, *Ergonomics of human-system interaction — Part 110: Dialogue principles*; and ISO 9241-171, *Ergonomics of human-system interaction — Part 171: Guidance on software accessibility*] address such design advice.
- Evaluating the design and the software product to assess how usable the software product will be in practice. ISO 13407 addresses such design advice.

Quantitative tests measure qualities such as task completion time and accuracy, while qualitative tests measure user attitudes and preferences.

The combination of quantitative and qualitative measurements selected for a specific usability test will depend on the purpose of the particular test and the point in the life cycle when the test occurs.

Once an analysis of usability requirements has been made and the appropriate testing methods have been selected, the documentation designer and developer should translate those requirements into usability goals for the documentation. The goals are the qualitative or quantitative targets, which are measured in usability tests. Figure 1 illustrates the process in the context of an electronic mail system.

Step 1. Define the user's objectives.

The user's objectives are:

- to find in the help menu the instructions for sending a message;
- to summarize the task in the user's own words;
- to send the message.

Step 2. Define the usability measures for those objectives.

The measures are:

- effectiveness (Is the right information found?);
- efficiency (How long does it take to find the information? Is the shortest search route or method used? Does the help text have to be re-read to be understood or remembered?);
- satisfaction (What are the user's attitudes towards the help?).

Step 3. Define acceptance criteria.

The criteria are:

- If the user finds the information within 20 seconds, then the structure and navigation are acceptable.
- If the user's summary of the task is correct, then the information is accurate and suitably clear.
- If the task could be performed correctly on the first try while following the instructions, then the help was acceptable, task-oriented, and complete.

Figure 1 — Sample process of defining the usability goals for an electronic mail system

Accessibility requirements extend usability requirements to ensure that the documentation is provided in suitable format for all those who may use the software product. The documentation should be provided in media and formats that allow its use by those with vision, hearing, or other physical limitation if they are able to use the software and to perform the tasks supported by the software.

EXAMPLE: The US government has published specific requirements for software accessibility, known as Section 508 of the Rehabilitation Act. Details of Section 508 may be found at the URL listed in *Annex H*.

Internationalization and national cultural requirements have a significant impact on the design of documentation, including schedules, costs, presentation formats, writing style, and usability testing. The designer should determine whether the application will be used by an international audience and in which countries. The answers to these questions will determine the translation requirements.

For software products to be made available in other countries using the same language as the source country, the documentation designer and developer should consider whether special localized versions of the documentation should be prepared for those countries, using appropriate local language, dialects, or conventions. Cultural issues should be considered, both in the software product and in the documentation, especially in examples. A national of the target country should check the user documentation to ensure that it is suitable for use in that country.

6.5 Interviewing technical contacts and other experts

Document designers and developers should conduct formal interviews as part of the process of gathering information and reaching consensus on design decisions. Interviews may be conducted through in-person meetings, telephone, instant messaging, e-mail, audio- or video-conferencing, Internet-based real-time chat tools, and other communication media used by the project team. Documentation designers and developers should first read the available software life cycle design documents, and then plan and conduct formal interviews with technical contacts and other experts to resolve questions. Interviews, whether conducted individually or in groups, should be scheduled as far in advance as practical. Frequent, unstructured, unscheduled meetings or questions may be perceived as interruptions by other team members. Documentation designers and developers should begin planning for interviews by identifying the information that they need to collect, along with the name or role of the persons or groups most likely able to provide the information, and then should carefully develop questions that will elicit the answers needed.

Group interviews often produce more information and resolve discrepancies in assumptions and approaches among members of the project team better than a series of individual interviews with the same persons. Documentation designers and developers should balance the schedules and time requirements of participants in determining which interview technique is best for a particular project. During the interview, no matter what medium is used to conduct it, those participating should have access to documents, prototypes, or software to reference or otherwise elicit information. Decisions and information gained from interviews and meetings should be published in meeting minutes to those participating and project managers, so that miscommunications may be promptly corrected and any continuing information gaps may be identified.

6.6 Project planning

The objective of documentation project planning is to identify the deliverable documents, specify the quality and usability standards they will meet, and define the project tasks, activities, schedules, resources, and cost. This clause describes factors in planning and controlling documentation projects from the viewpoint of the designers, writers, editors, artists, software developers, and others involved in user documentation development. Documentation designers and developers provide input to project managers for realistic configuration management, schedules, and cost estimates.

Documentation designers and developers should participate in meetings at which the requirements, objectives, and constraints of the project are identified and discussed.

NOTE: Documentation process planning and management will be addressed in a future International Standard (ISO/IEC 26511, *Software and systems engineering — User documentation requirements for managers*), the standard for the management of user documentation, which should be consulted by those responsible for developing documentation plans, controlling documentation development, or managing the content of user documentation so that it may be more easily reused.

The results of the documentation planning process shall be recorded in a documentation plan covering the documentation to be produced, and including individual plans (outlines or specifications) for each separate document item. The detailed documentation plans should be thoroughly reviewed along with the project plan. The documentation plan shall assign responsibility and authority for each major activity:

- technical accuracy of the documentation;
- usability of the documentation with the software product and vice versa;
- suitability of the documentation for the market;
- editorial quality of the documentation, including both text and illustrations;
- conformance to legal requirements;
- production of the documentation, including tools support;
- translation, localization, or customization;
- packaging and delivery of the software and the documentation;
- final approval of the documentation.

The audiences for the document plans are:

- the writing team;
- the reviewers;
- the people doing the analysis and planning;
- the product authority;
- the project managers who will be supervising the document development;
- customers who may need to know what the documentation will look like when it is complete;
- the people doing the translation, production, and usability testing;
- others in the organization responsible for marketing, training, and customer support.

The documentation plan should be approved with the rest of the product and shall be under change control procedures with the rest of the project plans.

NOTE: It is more difficult and more expensive to make major alterations to document structures or styles once development has started than it is to alter the design at the planning stage.

6.6.1 Quality management

The documentation development activities recommended in this International Standard should be carried out under the control of the quality management system (QMS) being used for the software product development. Users of this International Standard are recommended to operate a quality management system, which may be independently assessed for ISO 9000 compliance.

6.6.2 Version control and change control

A configuration management (CM) process shall be used to control the user documentation. The document designer should consider the CM policies that may influence the documentation design and affect its maintainability. These policies may include:

- the level at which version control is to apply; for example, whether it applies to a single file of information or to the information for a particular module, topic, object, or task;
- the landmarks within the project at which new versions are to be created; for example, after each set of tests;
- the method for controlling changes to each version;
- the method for applying version control to localized and customized versions;
- the back-up and archive procedures;
- the method for keeping records about the history of each version, to enable recreation of old versions if necessary;
- at what stage, if any, the documentation becomes part of the software for version control purposes.

Change control procedures for the project shall take account of the requirements of the documentation activities. Documentation developers should be members of any body responsible for approving changes. Documentation may be greatly affected by changes to the design of the software or the training or support plans. Therefore, the implications of such changes should be assessed before approval is given. When software product changes are made, documentation developers should be informed promptly, so that they work only with up-to-date information.

NOTE: Small changes to a system may cause major changes to documentation, although major changes to a system do not always require major changes to the documentation. For example, altering a system menu might be a small change to the software itself but might have major implications for the entire structure of a user manual. In contrast, completely revising the methods used within the software for some complex operation might be a major task in developing the software, but might not affect the user's view of the software at all, and therefore might require no changes to the documentation.

The same CM system used for the software development may support the documentation. This CM system will usually have some features appropriate for the documentation, particularly the on-screen elements.

NOTE: ISO/IEC TR 15846:1998, *Information technology — Software life cycle processes – Configuration management*, provides guidance on configuration management systems.

6.6.3 Availability of resources

During project planning, the designer and project manager should determine the availability of the resources needed for developing the documentation. The plan should include the time and effort to include acquisition of resources or services that are not already available in the organization. Resources include facilities, services, tools, and human resources, such as:

- the hardware and software for using a prototype or other versions of the software;
- a prototype or other versions of the software to be used by documentation developers to obtain information about the software being documented;
- the hardware and software for producing the documentation itself, for content management and storing master copies, and for translations;
- a laboratory for usability testing, and suitable space for other testing such as unit testing and functional testing;
- technical contacts who will supply information, discuss technical details with documentation developers, answer technical questions, and check drafts of the documentation;
- documentation developers or other staff who will editorially review drafts of the documentation;
- graphic designers and illustrators;
- printers and packaging vendors;
- usability staff;
- legal reviewers or contacts;
- translators.

6.6.4 Schedules

During the project implementation phase, the organization shall prepare a preliminary schedule for documentation activities. More detailed schedules for developing each separate document, as part of a master schedule for other parts of the project, shall be produced during the design and development phase. Documentation designers and developers, software developers, and the project manager should agree on the overall schedules for the project. Once the schedules are agreed upon, the team is committed to design documentation that may be prepared within the required time and cost.

When the project schedules are being defined, the documentation designer and developer should:

- remember that documentation development cannot be finished until after the software design is frozen;
- remember that the amount of time needed for the following activities may be significant and make adequate allowance for them in the project's schedules:
 - observing and using the software or prototypes of the software during the development stage;
 - learning how to use new tools;
 - obtaining information from technical contacts and having them check the accuracy of drafts;
 - creating the graphics and screen captures;
 - making draft documentation available for software product validation, field trials, and usability trials (documentation development schedules for preparing such drafts will affect the timing of these exercises and trials);
 - for each activity, particularly system tests, reviews, edits, and user tests, incorporating the results of the activities in corrected documentation; with subsequent retests and further revisions of the software and the documentation;
 - identifying technical work necessary to include the embedded documentation in the software product;

- obtaining legal reviews;
- having the documentation translated, if required;
- having the documentation printed and packaged, if required.

Once a date for delivering the application has been agreed upon, it should be applied in all phases of the planning. Changes in the software product and in planned delivery dates shall be promptly communicated to those concerned, and the impact of these changes on the documentation shall be evaluated and communicated to project management. If schedules need to be adjusted during the project, activities such as system tests, reviews, and usability tests should not be removed from the schedule to save time at the expense of quality.

6.6.5 Cost estimating

Documentation developers should provide input for estimates of the cost of developing the proposed documentation by including where relevant:

- documentation developer costs for development and for reviewing drafts;
- technical staff costs for briefing documentation developers and reviewing drafts;
- illustration costs;
- project management costs;
- editorial costs;
- tools support costs;
- training costs;
- expenses for support staff;
- equipment costs;
- materials costs;
- bulk copying and production costs (for example, copying and packaging media);
- costs of system validation tests, reviews, and user tests;
- evaluation costs;
- maintenance costs;
- costs of production, printing, delivery, or distribution to customers or users;
- translation costs for each language, which may include fees, travel, and communications.

NOTE 1: If documentation design is not taken into account in the design of the software product, the documentation might not be adequately concise, and hence documentation development costs might be higher than necessary.

NOTE 2: Similarly, if the product is to be translated or localized and those costs are not taken into account in the design of the software and the documentation, they may be significantly higher than necessary. The higher costs will be incurred for every translated or localized version.

NOTE 3: If future versions of the product are planned but not considered in the design of the documentation, the costs may be significantly higher. For example, if the document designer knows from the outset that a basic and enterprise

version of a product may be released, they may use publishing features such as variables for software product names and use those throughout the document during the original creation. Later, changing a single variable value will be much faster than searching for and replacing long and short software product names in one or more documents.

The documentation developer may assist in obtaining estimates of the costs of carrying out documentation development and production activities for a variety of documentation types to assist in the negotiation of documentation cost constraints.

6.6.6 Planning for localization and customization

The same factors considered in planning for the documentation project as a whole should be applied in planning localized or customized versions:

- Who is responsible for the process?
- Who will prepare the instructions for localization or customization, and when?
- Where will the localization or customization be carried out, and by whom?
- When and how will the product be delivered to the staff, and who will carry out the work?
- How will the localized and customized versions be tested?
- When and how will the localized or customized versions be delivered?
- Who is responsible for the quality of the localized or customized versions?
- Who is responsible for the usability of the localized or customized versions?

6.7 Documentation proposal

When documentation is to be produced by a separate organization (internal or external), the outcome of the objectives and planning activities may be a documentation proposal from the supplier to the acquirer. Figure 2 shows a sample contents list for a documentation proposal.

Documentation proposal contents

1. Executive summary

2. Overview of the objectives and requirements *(This section sets out the requirements specified for the project and briefly explains how they will be met.)*

- The product to be documented; future modifications; customization; security and safety concerns
- Target audiences; Internationalization and localization; accessibility concerns
- Critical tasks to be documented
- Legal requirements (including intellectual property issues)
- Organizational policies, procedures, and development tools affecting the project

3. Overview of the documentation to be provided

- Preliminary high-level design of the deliverable documentation
- International Standards and conventions to be followed
- Documentation delivery and viewing constraints

4. Documentation development and review processes *(This section may include separate sections for localization, translation, or customization projects.)*

- Process overview
- Acquirer's resources and responsibilities
- Supplier's resources and responsibilities
- Preliminary schedule
- Tools and systems to be used
- Project risks

5. Business proposal

- Basis of estimate (comparison of the project to previous schedules, costs, and resources)
- Ownership of intellectual property and delivered documentation
- Cost and payment terms

Figure 2 — Sample contents list for a documentation proposal

7 Analysis and design

Analysis and design are the first tasks in the documentation design and development process. Analysis is the process of developing detailed requirements and acceptance criteria for the documentation. Documentation shall be based on audience and task analyses. Taking these analyses into consideration, design is the process of specifying the structure, content, format, and style to be used in the documentation.

7.1 Audience and task analysis

7.1.1 Audience analysis

The designer shall list the intended types of users of the software product and classify users into audiences. Each audience consists of a group of users whose work tasks and use of the application are similar. The classification should consider:

- the users' background, experience, and training;
- the language familiar to the users;
- the way that the application will be used;
- the users' learning stages (for example, how much experience they have with this application);
- the frequency of use (some users use an application or feature frequently, some infrequently);
- the users' working environments (do the users do most of their work at a desk, or up a telephone pole?).

NOTE: An analysis of the tasks performed by members of each audience will help identify their information requirements (see *Clause 7.1.3*). However, the above factors will influence the level of detail required, the ratio of conceptual to procedural information presented, as well as the best way to present and organize the information.

Figure 3 shows a sample list of audiences for part of an order fulfillment system:

Call centre operator
Call centre supervisor
Webmaster
Warehouse picking staff
Warehouse manager
Finance manager
Accounts manager
Accounts payable clerk
Accounts receivable clerk

Figure 3 — Sample list of audiences for part of an order fulfillment system

The documentation designer should use both a “bottom-up” and a “top-down” approach to check that all user types have been considered:

- Bottom-up—The documentation designer should consider who will use the software product and determine all the types of user. For some applications, the audiences may be identified by their job titles, such as accountant. For other applications, audiences may have to be identified by the role they are performing, such as letter writer.
- Top-down—The documentation designer should consider the whole organization or the total functionality of the application, and break this down until a set of audiences or roles is reached. This set of audiences may be used to check the results of the bottom-up analysis.

User roles may not correspond to people in a 1:1 relationship; the same person might have more than one role: for example, salesperson and inventory taker.

Audiences may be grouped into a hierarchy, so that individual documents may be targeted at several audiences. The documentation designer should use the hierarchy to group together audiences who will have the same type of interaction with the software. The documentation designer should not (necessarily) reproduce the user organization's organization chart. Figure 4 is an example of such a hierarchy.

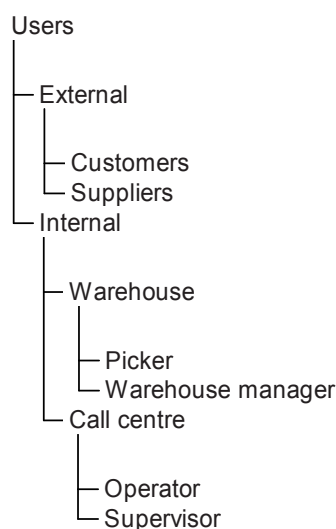


Figure 4 — Sample of an audience hierarchy

Audiences who are similar in other respects require different types of documentation according to how long they have been using the application or how often they use it. For example:

- learning to use (explanation of concepts and tutorials on procedures);
- using occasionally or infrequently (procedural instructions);
- using regularly (quick-reference cards);
- performing advanced or complex tasks (reference material and additional procedures).

If the software product being developed is a consumer software package, it may not be possible to gather useful information about typical users' jobs and background, for example, because their jobs and skills are varied. In this case, the designer should concentrate on the tasks that the users will carry out and the learning stages through which the user will go. Where there might be a wide variety of types of potential user, with a wide variety of experience, skills, and knowledge, the documentation should be adequate for use by the least experienced of the expected users.

The documentation designer should collect details of users' working environments. These details are one factor in deciding the most convenient media for presenting information. For example, if the software is to be used in a storeroom or warehouse:

- Will printed documents have to be carried around?
- Will there be convenient places to store them?
- Will users have the equipment to view the information on-screen?

If you expect the documentation to be used by a wide range of users with varying levels of experience and usage frequency, it is important to "layer" the information carefully. Layering ensures that the material is presented in such a way that all users may find the information that they need, at the level of detail that they need. In printed documentation, layering usually involves presenting information in progressively more detail, using clear headings to enable users to select the right information. In online documentation, layering may be achieved by various methods, including the use of hypertext links from overview-level items to items containing detailed information.

7.1.2 Audience profiles

For each type of user, the documentation designer should draw up an audience profile that records the pertinent information. This profile will be useful both in planning and as a guide for writers and illustrators. The audience profile may be recorded either in absolute terms (for example, special qualifications of one audience) or in a comparison of users' experience to the norm. Figure 5 illustrates the types of information that might be included in an audience profile.

Audience: Ticket Agent	
Background	Ticket agents have knowledge of the travel industry, customers' needs and concerns, and other computer booking systems.
Languages	Ticket agents may have a command of English, but not necessarily as their first language.
Use of the application	Ticket agents use the application while they are on the telephone to customers or when the customer is present, and make ticket bookings immediately.

Learning stage	All ticket agents will have attended a one-day training course; consequently, no users are novices. There is at least one expert user in each office all the time.
Frequency of use	Ticket agents frequently use the application all the time throughout an eight-hour shift (with breaks).
Working environment	Offices may be noisy and very busy. Ticket agents sit at desks to work. Agents have access to a computer throughout the shift but may share printers. Offices usually have a photocopier, fax machine, and scanner. Ticket agents use headsets for the telephone. Not all offices will have: • shelf space to hold a small library of books; • drawer space in the desks; • wall space available for displaying wall charts.

Figure 5 — Sample audience profile

7.1.3 Task analysis

The documentation designer should collect information about what users do, if possible by asking users questions, or by observing users and recording what they do. If this cannot be done (for example, if the software is still under development), then the documentation designer and developer should consider the tasks, or use other sources such as use-case design documents.

Figure 6 shows a sample task list for an electronic mail system.

Check for incoming mail. Send or forward a message. Print a message. Attach a file. Read an attachment. Create a mail folder. Move mail to a folder.

Figure 6 — Sample task list for an electronic mail system

The documentation designer should consider the following information about each task:

- why the task is carried out;
- how frequently the task is carried out (to help determine whether users will remember how to do it);
- how the user is likely to be operating, for example, doing tasks that take several hours, or working in a real-time situation where transactions have to be carried out while a customer is present;
- what discretion the user has in how or when the task is carried out;
- in what sequence the user carries out the set of tasks;
- what the prerequisites for the task are;
- how fault-tolerant the task is; that is, how important it is that the user carries out the task correctly;
- what the consequences of the task being completed or not completed are.

The designer should group related or similar tasks, or tasks that involve similar steps, in a hierarchy as illustrated in Figure 7. The designer should record the details of each task in a task profile.

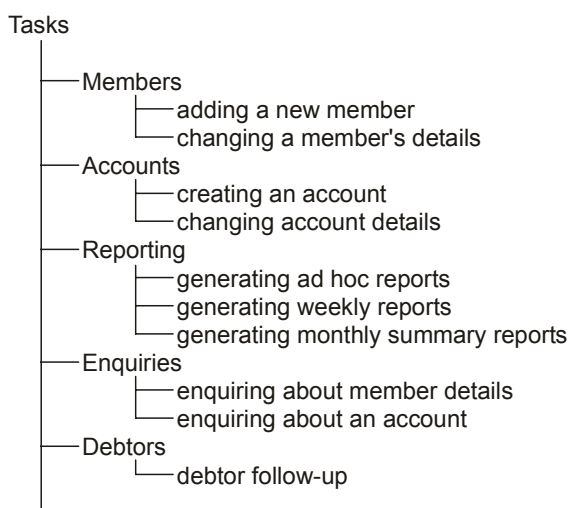


Figure 7 — Sample task hierarchy

The documentation designer should document which audiences carry out which tasks. The designer should consider using a matrix, as illustrated in Table 1. This example shows which types of user will be carrying out a set of security-related tasks and what their learning stages will be. As a result of this audience and task analysis, the documentation could be structured into a guide for the functional analyst, a guide for the security officer, and a quick reference (help topic) for users on changing their passwords. Common procedures could be written for the task of viewing the security log to be included in both guides.

Table 1 — Audience mapping matrix

Task	Audience								
	DIC	SUP	MAN	AUD	ACC	FC	SO	FA	SA
Adding a new user							N		
Deleting a user							N		
Giving a user permission to access a function								N	
Withdrawing a user's permission to access a function								N	
Viewing a user's permission to access functions								N	
Changing your password	O	O	O	O	O	O	O	O	O
Changing another user's password							NE		
Viewing the security log							N	N	
Investigating security breaches							NE	NE	
Printing security reports							NE	NE	

Legend to Table 1

Key to audiences		Key to learning stages	
DIC	Data input clerk	O	Occasional use
SUP	Supervisor	N	Normal use
MAN	Manager	E	Exploiting advanced features
AUD	Auditor		
ACC	Accountant		
FC	Financial controller		
SO	Security officer		
FA	Functional analyst		
SA	System administrator		

7.2 User documentation design

The user documentation designer shall perform two principal activities:

- use the audience and task analysis to determine what information needs to be provided (the documentation content) and what will not be provided;
- structure the content into necessary items in the documentation set, and determine the structure for each documentation product.

Depending on the software in question, a number of the items in this clause may apply more to a documentation set than to a particular document. For example, one piece of software may have a tutorial for end users, a manual for administrators, a reference for the application program interface, online help, a quick reference guide, and so on.

Subsequent clauses of this International Standard address the structure and content required for user documentation. For guidance on deciding whether information should be delivered on the screen or on paper, see *Clause 12.2*.

7.2.1 Designing for use of content

The documentation designer should prepare a list of the topics, in addition to the relevant access and structural information. The major factors that should influence decisions about where and how information is provided to the user are:

- audience and task profiles;
- information characteristics;
- the environment in which the information will be used;
- convenience to users;
- the range of technical facilities, including resources and the software product, available for developing, delivering, and viewing documentation;
- cost of delivery and maintainability.

The documentation designer should select a measure of how much information there will be, and use it to make sensible choices about delivery methods and delivery media. Volume estimates will vary according to the type of information. It may not be possible to find the volume of the information precisely, but the designer should have a concept of the scale of the documentation. The documentation designer should define the preliminary structure of each document using a draft table of contents, a topic list, or a list of web pages. To specify the size of each document, the documentation designer should give a count of the expected number of pages or topics. The documentation designer should include in the list navigation pages or topics, such as the contents list itself, indexes, glossary, and bibliography. The following techniques will be helpful.

- The designer may count identifiable items, for example, icons, tasks, dialog boxes, transactions.
- The designer may assess how much information is needed about each item, for example, the average and maximum number of steps for a task, or the approximate number of words needed about a concept.
- The designer may prepare a sample for one item and use its size as a basis for calculations.
- The designer may refer to existing documentation for comparable users, tasks, and applications, and assess from that how much information will be needed for the current application.

7.2.2 Designing formats

Consistent documentation is more easily used, because the user can focus on the content rather than the format. The organization shall set conventions for the whole product, project, or organization for entire documents and document sets, topics, chapters, front matter, back matter, prefaces, technical terminology, icons and symbols, navigational controls, and system messages.

For details of documentation formatting, refer to *Clause 12*.

8 Development and review

Although prototypes or samples of documentation may be developed as part of the documentation design process, the full development and review process should begin after the design of the documentation is finished. With a complete documentation plan, the documentation developer should follow the writing and illustration styles specified in the detailed document plan, using the specified tools and methods and including only the information specified in the detailed document plan.

Documentation development consists of writing the relevant texts, preparing the illustrations, and applying the necessary software elements to on-screen documentation to create a working system. In the documentation development process, the documentation developer prepares master versions of technically accurate documentation topics (also called objects or chunks) for assembly into the documentation products as specified in the documentation plans.

NOTE: The process of developing individual documents is an iterative one. If changes are made to the software, then changes might be needed in the documentation plan and the documentation.

8.1 Prototypes and drafts

Prototypes may be used to validate documentation designs for each type of document. Prototypes may be developed in detail and used to test that the documentation approach will provide users with what they need within the usability objectives. Documentation prototypes may be developed in conjunction with software prototypes. Prototypes may also be prepared for combinations of equipment, and for languages or localized versions, which will be used for the finished application. For each prototype, the documentation developer should consider the following questions:

- What is its purpose? What information is intended to be collected based on testing or trials of the prototype?
- What is its scope? What part of the application will be covered, and what types of documentation will be included?
- How will the tests and trials be carried out, recorded, and used in subsequent phases?

Prototypes are also useful for on-screen documentation to check that technology issues are resolved before the documentation design is finalized.

To avoid problems arising later in the project, at the first draft stage, the documentation developer should:

- try out new documentation-related methods and tools being used in the project, for example for creating or capturing illustrations or for output of content in various media;
- produce a sample of the text and illustrations using the final production method;
- for online documentation, check the context-linking of topics to application features; that is, make sure that the context-mapping technology is working correctly and that the correct context identifiers are being used.

NOTE 1: In planning printed documents, the documentation developer should decide for each document separately whether or not prototypes and drafts should be presented using the screen or page layout to be used for the finished work. Using the layout for the finished document helps those commenting on drafts see what the finished documentation will look like, but might lead reviewers to concentrate on page layout features instead of the technical content of the documents.

The documentation should be drafted when:

- outlines in the detailed documentation plans have been authorized;
- functional requirements documents have been accepted;
- sufficient stable information about the software is available.

Before drafting documentation, the corresponding part of the application software should exist, or there should be an agreed set of requirements and nomenclature for the software. If the information needed is not available by the date specified in the schedule, the documentation developer should advise the project manager to review the project schedule.

Before the documentation developer drafts task instructions, the project team should agree on the names of the application features that the tasks use, such as the names of menu selections, icons, and navigation features. The documentation developer should prepare a project glossary and apply it consistently. Clarity and simplicity of style are particularly important in documents that will be translated. The documentation developer should avoid slang, jargon, and humor, and use similes and analogies with care. In addition, the documentation developer should be aware that idioms and metaphors might not translate correctly, might have no equivalent in the target language, or could actually be offensive in another culture.

NOTE 2: For documentation that will be translated, the project team should agree which terms in the project glossary will remain in the source language and consistent terminology should be used for terms translated into the target language.

The documentation developer should ensure that the information contains everything users need and nothing that is irrelevant. The documentation developer should use the software to check the tasks being documented. The drafts should be technically accurate, based on the following sources of technical information:

- documentation design information;
- system design information;
- SME or software developer's information;
- personal experiences of using the system or a prototype of it.

NOTE 3: A member of the documentation team may be the first person to study the working software in detail from a user's perspective. The documentation developer's use of the software may provide a rigorous test of a system and give valuable feedback to the development team, for example by identifying inconsistencies.

NOTE 4: In planning, the documentation designer should decide whether it will be useful to provide reviewers with a draft or partial index for the first draft of each document so that they can find topics in the draft and review the format of the index with the rest of the draft.

8.1.1 CM during development

The organization responsible for documentation development shall ensure that the requirements of the CM system for the project are enforced. Approved documentation versions shall be maintained securely and separately from copies checked out for development. Each draft issued for review shall be uniquely identified. Back-up copies of master documentation should be stored securely and separately from the systems being used for documentation development.

8.1.2 Development of translated and localized documentation

While it is simpler to translate or adapt complete, approved documentation, documentation must often be released in multiple languages simultaneously to support software product releases. For some software products that are to be exported, there might be a legal requirement for the documentation to be translated.

Translations should be made by native speakers of the target language, not by native speakers of the source language. Localized versions may be needed for variations in the target language: for instance, European Spanish versus Mexican Spanish, and UK English versus US English.

As a first step in the translation process, the project glossary (list of terms and their definitions) shall be translated. The documentation shall be translated only after the translated list of terms has been approved. This is especially important when more than one translator is involved for a language (the structure of the glossary is described in *Clause 11.12*).

If inaccuracies or ambiguities in the source text are found during the translation process, the documentation developers should revise the source at the earliest opportunity.

Translated and localized documentation should be reviewed and tested in the same way as the base version.

8.2 Evaluation of documentation

The purpose of documentation evaluation is to ensure that documentation is fit for purpose. Documentation evaluations are performed throughout the document's development, production, and maintenance, from first draft to revision, for reuse or consideration of enhancements. The evaluation method selected will depend on a variety of factors including:

- the reasons for carrying out the evaluation;
- the phase in the life cycle at which the evaluation is being carried out;
- the resources available;
- the time available;
- the amount of information available about the documentation;
- the availability of users with the appropriate ranges of experience and skill;
- the availability of experts in documentation design;
- the availability of experts in usability.

Different methods may be used at different phases in the development, after the documentation is complete, and when the documentation has been used for a specified period. A combination of methods is likely to be appropriate in each case.

This clause describes the processes of documentation evaluation in two forms: review and testing.

The documentation review process includes reviews of:

- documentation structure, format, and style compared to plans, requirements, and established standards;
- scope and level of detail of the information presented;
- range of topics covered;
- technical accuracy;
- consistency with the product;
- safety (provision of critical information to protect against hazards or errors).

The documentation test process includes:

- operational testing to ensure that the documentation performs correctly and responsively (for example, on-screen documentation is appropriately linked to the software topic, and navigation is consistent and as expected; the index in printed documentation has accurate references);
- usability testing of documentation with the software product, to determine whether the intended audience can perform their tasks with the aid of the documentation.

Documentation test requirements shall be specific and measurable.

After documentation is released, evaluation continues in the form of feedback from users (through comment forms and messages) and from independent software product review magazines and reports. Surveys and interviews may also be useful in gathering users' and customers' opinions and attitudes on released documentation. Problem reports gathered by trainers, sales staff, and customer service also indicate where documentation or software products need improvement.

NOTE: Documentation is also included in evaluation of the software product, as in ISO/IEC CD 25012, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Data quality model*. The ISO/IEC 9126 series (*Software engineering — Product quality*) may also be used for this purpose. The ISO/IEC 14598 series (*Software engineering — Product evaluation*) addresses software product evaluation.

Documentation evaluation shall include the following four activities:

- **Plan:** The documentation designer and developer should identify the requirements for acceptable performance or quality, and prepare for the evaluation exercise by specifying when the test will be held, what resources are needed, how the evaluation will be carried out (test scenarios and scripts), how the results will be measured and recorded, how the results will be analyzed, and what the pass/fail criteria for the tests will be.
- **Do:** The documentation designer and developer should evaluate the documentation against the requirements, taking measured results and recording them. Members of the project team normally carry out tests during development following test scripts that are drawn up in advance to ensure that the tests are systematic and complete.
- **Check:** The documentation designer and developer should analyze and report the results of the evaluation, recommending next steps.
- **Act:** The documentation designer and developer should revise the documentation, advise changes to the software product based on the test results and recommendations, and determine whether further evaluation cycles are needed.

To produce acceptable results, revisions may entail changes in the project schedule and documentation plan, or even changes in the requirements. For example, documentation of advanced functions might be deferred until a later project phase, or product release dates might be deferred until acceptable documentation is produced.

8.2.1 Other roles in evaluation of documentation quality

Documentation evaluation shall be based on the required features and qualities. Evaluation of documentation quality depends on the recognition of various perspectives for acceptability. Ultimate acceptance comes from the end-users, but unless others find the documentation to be of acceptable quality, it will never reach the users. Those other perspectives include:

- **Manager's view.** Managers may be more concerned with overall quality than with specific quality characteristics. They may assign different weights to certain characteristics to reflect the business needs of the organization, comparing the documentation to what is commercially available in the market and what is less costly to produce. Managers should be aware that the quality of documentation may appreciably affect costs for customer support and future sales.
- **Technical editor's view.** The technical editor assesses overall document quality and consistency and works to ensure that documentation fulfills the vision and plans that were laid out during the analysis phase.
- **Developer's view.** Software developers may be concerned with how the software product operates in its novel or advanced functions, more than with how it supports the users' tasks.

NOTE: However, developers are very close to the software product and are more sophisticated users.

- **Maintainer's view.** Those who will have to maintain the documentation systems will have special requirements for quality in addition to those of other documentation developers. They will be concerned with, for example, the simplicity and clarity of the documentation structures, the ease with which new versions of the documentation may be created, and portability to new technology for content management and production.
- **User's view.** Users are likely to measure quality in terms of the inclusion of accurate and relevant information, as well as ease of finding the required information and applying it.

8.2.2 Documentation review procedures

Documentation reviews should precede usability testing to improve documentation quality, reducing the number of flaws and defects remaining to be identified in testing and thus the amount of rework and retest required at a late stage. Documentation drafts should be reviewed for the following:

- **Technical accuracy.** The approval authority identified in the documentation plan has responsibility for the technical accuracy of the documentation for the software product and for resolving conflicting comments from technical contacts.

NOTE 1: Technical accuracy of instructional procedures is better verified by testing the documentation with the software than by reviews. Technical reviews of embedded documentation (such as pop-up help) are best performed in conjunction with software product testing.

- **Safety.** An expert should review the documentation to ensure that it contains warnings, cautions, and appropriate avoidance and recovery instructions for risks associated with use of the software product. The reviewers shall verify that warnings and cautions are appropriately placed and worded to be noticeable and understandable by the intended users.
- **Ease of understanding.** At least one person who is not familiar with the software product should carry out the checks for ease of understanding. The person checking drafts should be alert to potential misreadings and misunderstandings, and should highlight them.
- **Conformance and consistency.** Documentation developers or editors should check that the documents conform to the requirements of selected standards and organizational policies and style guides, follow the documentation and design plans, and are consistent with each other in appearance and terminology. The reviewers should check for consistency and accuracy in references or links to other parts of the document or document set. This level of assessment will not guarantee that the documentation will satisfy its users, but should ensure that no user will find the documentation totally unsatisfactory.
- **Completeness.** The documentation should include or refer to the information that users need. If possible, at least one person who is not familiar with the software product should carry out the checks. (See *Clause 11.1*.) The total set of documentation, printed and on-screen, should be reviewed, including the items that will be integrated with the software. The documentation should be reviewed together, or be reviewed by the same people, so that it may be checked for consistency and completeness.
- **Editorial consistency and correctness.** An editor should check near-final drafts for spelling, grammar, punctuation, editorial standards, and all documentation structure, format, and style requirements.
- **Security and legal accuracy.** The organization's security group and legal counsel should check near-final drafts to ensure inclusion of the correct notices for protection of sensitive and proprietary information, compliance with legal requirements, and proper handling of trademarks. If possible, an intellectual property law (IPL) attorney should perform this check.

The documentation developer and project manager should determine the review participants and procedures. Reviewers may be peers, editors, technical experts, trainers, managers, or customers. Reviewers should be selected based on their expertise, familiarity with the requirements and standards, and ability to provide thorough and usable comments and corrections. It is rare for reviewers to focus on all aspects of documentation at one time. Separate reviews or reviewers should evaluate the documentation's organization; technical accuracy; suitability for the intended audience; and consistency of grammar, style, and format.