INTERNATIONAL STANDARD

ISO/IEC/IEEE 26514

First edition
2022-01

Systems and software engineering — Design and development of information for users

Ingénierie du logiciel et des systèmes — Conception et développement d'informations pour les utilisateurs

Reference number
ISO/IEC/IEEE 26514:2022(E)

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

Institute of Electrical and Electronics Engineers, Inc
3 Park Avenue, New York
NY 10016-5997, USA

Email: stds.ipr@ieee.org
Website: www.ieee.org

Published in Switzerland

Contents

Page

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.

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

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see https://patents.iec.ch).

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

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

ISO/IEC/IEEE 26514 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Systems and software engineering*, in cooperation with the Systems and Software Engineering Standards Committee of the IEEE Computer Society, under the Partner Standards Development Organization cooperation agreement between ISO and IEEE.

This first edition cancels and replaces ISO/IEC 26514:2008, which has been technically revised.

The main changes are as follows:

— increased emphasis on designing and developing information for users of software;

— use of IEC/IEEE 82079-1 as a normative reference for information for use;

— addition of subclauses regarding application programming interfaces (API) and chatbots.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

Anyone who uses software designed to help users perform particular tasks or handle particular types of problems needs accurate information about how the software helps the user accomplish a task. The information for users 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 information for users 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 little separate explanation 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 mechanisms are discernible by users, but only through laborious testing. For these reasons and more, information for users remains an essential component of usable software products.

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

This document provides specific requirements for information for users of software products, based on the requirements applicable to all types of products in IEC/IEEE 82079-1. It focuses on the parts of the information management processes most applicable for information designers and information developers.

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

This document was developed to assist users of ISO/IEC/IEEE 12207 to design and develop information for users as part of the software life cycle processes. It defines the information-development process from the information developer's standpoint.

Other documents (ISO/IEC/IEEE 26511, ISO/IEC/IEEE 26512, ISO/IEC/IEEE 26513, ISO/IEC/IEEE 26515, and ISO/IEC/IEEE 26531) address the information management process from the viewpoints of managers, acquirers and suppliers, reviewers and testers, participants in agile development work, and content managers.

In addition to defining a standard process, this document also covers the information product. This document specifies the structure, content, and format for information for users, and also provides informative guidance for the style of such information.

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

This document is independent of the software tools that may be used to produce information for users, and applies to both printed and onscreen information, as well as information presented by other methods such as animation or video. Much of its guidance is applicable to information for users of systems including hardware as well as software.

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

The order of clauses in this document does not imply that the information for users 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 9.

Systems and software engineering — Design and development of information for users

1 Scope

This document covers the development process for designers and developers of information for users of software. It describes how to establish what information users need, how to determine the way in which that information should be presented, and how to prepare the information and make it available. It is not limited to the design and development stage of the life cycle, but includes information on design throughout the life cycle, such as design strategy and maintaining a design.

This document provides requirements for the structure, information content, and format of information for users of software.

This document can be applied to developing the following types of information, although it does not cover all aspects of them:

- information for users 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;
- maintenance information describing the internal operation of systems software;
- information for users incorporated into the user interface itself.

This document is applicable to information architects and information developers, including a variety of specialists:

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

This document is also a reference for those with other roles and interests in the process of developing information for users:

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

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

IEC/IEEE 82079-1:2019, *Preparation of information for use (instructions for use) of products – Part 1: Principles and general requirements*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

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

ISO, IEC, and IEEE maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/

— IEC Electropedia: available at https://www.electropedia.org/

— IEEE Standards Dictionary Online: available at https://dictionary.ieee.org

NOTE For additional terms and definitions in the field of systems and software engineering, see ISO/IEC/IEEE 24765, which is published periodically as a “snapshot” of the SEVOCAB (Systems and software Engineering Vocabulary) database and is publicly accessible at https://www.computer.org/sevocab.

3.1.1
accessibility
extent to which products, systems, services, environments and facilities can be used by people from a population with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use

Note 1 to entry: Although “accessibility” typically addresses *users* (3.1.54) who have disabilities, the concept is not limited to disability issues.

[SOURCE: ISO/IEC 25064:2013, 4.1, modified — The original note 1 to entry has been removed and replaced by a new one.]

3.1.2
accuracy
quality of information that it is correct and consistent with a *software product* (3.1.47)

3.1.3
action
element of a *step* (3.1.48) that a *user* (3.1.54) performs during a *procedure* (3.1.39)

3.1.4
active area
<*onscreen information for users* (3.1.36)> area that responds to *user* (3.1.54) control or manipulation

EXAMPLE A hot-spot on a graphic, a hyperlink in text, a button in a screen display.

3.1.5
analysis
investigation and collection task of *development* (3.1.18) that aims to specify types of *users* (3.1.54) and needed information

3.1.6
application programming interface
API
set of *functions* (3.1.23), protocols, parameters, and objects of different formats, used to create *software* (3.1.46) that interfaces with the features or data of an external system or service

EXAMPLE The interface to a suite of service subroutines, a set of dedicated URLs that return data in response, or a suite of *commands* (3.1.11) that can be issued to a physical device such as a robot on an assembly line.

Note 1 to entry: *Information for users* (3.1.29) of an API is of two main types: *reference information* (3.1.43) (which contains information about all elements of the API) and developer guide (which explains how to use the API).

Note 2 to entry: APIs can take several forms. In general terms, an API is a set of clearly defined methods of communication among various components. An API specifies the information and methods that are needed to communicate with another application.

3.1.7
audience
category of *users* (3.1.54) sharing the same or similar characteristics and needs (for example, purpose in using the *information for users* (3.1.29), tasks, education level, abilities, training, and experience) that determine the content, structure, and use of the intended information

Note 1 to entry: There may be different audiences for information for users (for example, management, data entry, maintenance, engineering, business professionals).

[SOURCE: ISO/IEC/IEEE 23026:2015, 4.2, modified — "documentation" has been replaced by "information for users"; in note 1 to entry, "engineering, business professionals" has been added.]

3.1.8
caution
advisory information that states that performing some *action* (3.1.3) can result in minor or moderate injury or lead to consequences that are unwanted or undefined, such as loss of data or an equipment problem

Note 1 to entry: See also *danger* (3.1.16) and *warning* (3.1.56).

3.1.9
change control
actions (3.1.3) taken to identify, document, review, and authorize changes to a product that is being developed

Note 1 to entry: The *procedures* (3.1.39) confirm the validity of changes, that the effects on other items are examined, and that those people concerned with the *development* (3.1.18) are notified of the changes.

3.1.10
computer-aided design
CAD
use of a computer to design a device or a system, display it on a computer monitor or printer, simulate its operation, and provide statistics on its performance

3.1.11
command
expression that can be input to a computer system to initiate an *action* (3.1.3) or affect the execution of a computer program

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.638, modified — The EXAMPLE has been removed.]

3.1.12
conceptual information
explanations and descriptions which enable the *audience* (3.1.7) to understand the product's operating principles in order to perform required tasks

3.1.13
configuration management
CM
technical and organizational activities, comprising configuration identification, control, status accounting and auditing

[SOURCE: IEEE 828-2012]

3.1.14
context-sensitive help
type of *onscreen information for users* (3.1.36) in which the material that is displayed depends upon the current status of the *software* (3.1.46) and the progress of the *user's* (3.1.54) task

3.1.15
customize
adapt a *software* (3.1.46) or information product to the needs of a particular *audience* (3.1.7)

3.1.16
danger
hazardous situation, which if not avoided, can result in death or serious injury

Note 1 to entry: See also *caution* (3.1.8) and *warning* (3.1.56).

[SOURCE: ISO/IEC/IEEE 26513:2017, 3.8]

3.1.17
design
stage of information *development* (3.1.18) that is concerned with determining what *information for users* (3.1.29) is to be provided in a product and what is the nature of that information

3.1.18
development
activity of preparing *information for users* (3.1.29) after it has been designed

3.1.19
document
uniquely identified unit of information for human use

[SOURCE: ISO/IEC/IEEE 15289:2019, 3.1.10, modified — The EXAMPLE and note 1 to entry have been removed.]

3.1.20
document set
documentation (3.1.21) that has been segmented into separately identified volumes or products for ease of distribution or use

[SOURCE: ISO/IEC/IEEE 26513:2017, 3.11, modified — "collection of" has been removed.]

3.1.21
documentation
collection of documents related to a given subject

[SOURCE: IEC/IEEE 82079-1:2019, 3.11]

3.1.22
embedded information for users
information for users (3.1.29) that is accessed as an integral part of *software* (3.1.46)

EXAMPLE Pop-up help and help text on a screen.

Note 1 to entry: See also *onscreen information for users* (3.1.36) and *printed information for users* (3.1.38).

3.1.23
function
defined objective or characteristic *action* (3.1.3) of a system or component

3.1.24
icon
graphic displayed on the screen that represents a *function* (3.1.23)

[SOURCE: ISO/IEC 11581-1:2000, 4.7, modified — "of a visual display" and "of the computer system" have been removed.]

3.1.25
illustration
graphic element set apart from the main body of text and normally cited within the main text

Note 1 to entry: The term “illustration” is used as the generic term for tables, figures, exhibits, screen captures, flow charts, diagrams, drawings, *icons* (3.1.24), and other types of graphics.

3.1.26
information architect
person who develops the structure of an information space and the semantics for accessing information on tasks, system *functions* (3.1.23) and features, and other information

[SOURCE: ISO/IEC/IEEE 26513:2017, 3.20, modified — "required task objects, system objects" has been replaced by "information on tasks, system functions and features".]

3.1.27
information architecture
structure of an information space and the semantics for accessing information on tasks, system *functions* (3.1.23) and features, and other information

3.1.28
information developer
person who prepares content for *information for users* (3.1.29)

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.8]

3.1.29
information for users
information provided by the supplier that provides the target audience with concepts, *procedures* (3.1.39) and reference material for the safe, effective, and efficient use of a supported product during its life cycle

EXAMPLE Printed manuals, onscreen information, and stand-alone online help are examples of information for users.

Note 1 to entry: Throughout this document, the term “information for users” refers to information for *users* (3.1.54) of *software* (3.1.46).

Note 2 to entry: It can be provided separately or embedded in the product or both.

Note 3 to entry: The term “information for users” in this document is intended as a synonym for “information for use” as used in IEC/IEEE 82079-1 and is defined as such, although it is recognised that the respective ideas, while similar, are not identical.

[SOURCE: IEC/IEEE 82079-1:2019, 3.17, modified — The preferred term has been changed from "information for use" to "information for users"; the admitted term "instructions for use" has been removed; the original EXAMPLE and notes to entry have been removed and replaced by new ones.]

3.1.30
instructional information
information that explains how to use a product, system, or service to perform tasks

3.1.31
internationalization
process (3.1.40) of developing information so that it is suitable for an international *audience* (3.1.7)

Note 1 to entry: See also *localization* (3.1.32).

3.1.32
localization
creation of a national or specific regional version of a product and its *information for users* (3.1.29)

Note 1 to entry: Localization may be carried out separately from the translation *process* (3.1.40).

Note 2 to entry: See also *internationalization* (3.1.31).

3.1.33
menu
list displayed on a screen showing available *functions* (3.1.23) from which the *user* (3.1.54) can select an *action* (3.1.3) to be initiated

3.1.34
minimalism
principle that *information for users* (3.1.29) includes critical information and the least amount of other information needed to be complete

[SOURCE: IEC/IEEE 82079-1:2019, 3.25, modified — "information for use" has been replaced by "information for users".]

3.1.35
navigation
act of accessing *information for users* (3.1.29) and viewing different *topics* (3.1.51)

3.1.36
onscreen information for users
information for users (3.1.29) that is intended to be read on the screen by the *user* (3.1.54) while using the *software* (3.1.46)

EXAMPLE Pop-up help and help text on a screen.

Note 1 to entry: See also *embedded information for users* (3.1.22) and *printed information for users* (3.1.38).

3.1.37
picture
illustration (3.1.25) that shows the actual appearance of physical objects

EXAMPLE Photographs and drawings.

3.1.38
printed information for users
information for users (3.1.29) that is either provided in printed form, or provided in electronic form for the customer or *user* (3.1.54) to print

Note 1 to entry: See also *embedded information for users* (3.1.22) and *onscreen information for users* (3.1.36).

3.1.39
procedure
ordered series of *steps* (3.1.48) that specify how to perform a task

3.1.40
process
set of interrelated or interacting activities which transforms inputs into outputs

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.33]

3.1.41
project
endeavour with defined start and finish criteria undertaken to create a product or service in accordance with specified resources and requirements

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.37, modified — Note 1 to entry has been removed.]

3.1.42
project manager
person with overall responsibility for the management and running of a *project* (3.1.41)

3.1.43
reference information
information that is intended to provide quick access to specific details for *users* (3.1.54) who are generally familiar with the product's *functions* (3.1.23)

3.1.44
secondary window
window (3.1.57) that contains information that depends on information in another window (the primary window)

Note 1 to entry: The information in the secondary window supplements the information in the primary window.

3.1.45
signpost
text, symbol, or small graphic that helps the *user* (3.1.54) identify where particular types of information are located or where the information in the current display fits into the *information for users* (3.1.29) as a whole

Note 1 to entry: Information of different types may be indicated by symbols or graphics of different types.

3.1.46
software
all or part of the programs which *process* (3.1.40) or support the processing of digital information

Note 1 to entry: For the purposes of this document, the term “software” does not include *onscreen information for users* (3.1.36).

[SOURCE: ISO/IEC 19770-1:2017, 3.49, modified — The original notes 1 and 2 to entry have been replaced by a new note 1 to entry.]

3.1.47
software product
set of computer programs, *procedures* (3.1.39), and possibly associated *information for users* (3.1.29) and data

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.54, modified — Note 1 to entry has been removed.]

3.1.48
step
element (numbered list item) in a *procedure* (3.1.39) that tells a *user* (3.1.54) to perform an *action* (3.1.3) (or actions)

Note 1 to entry: Responses by the *software* (3.1.46) are not considered to be steps.

3.1.49
style
set of language-specific editorial conventions covering grammar, terminology, punctuation, capitalization, and word choice

3.1.50
subject-matter expert
SME
person responsible for providing technical information or for checking the technical *accuracy* (3.1.2) of drafts of *information for users* (3.1.29)

3.1.51
topic
coherent part of an information product with a heading that deals with a single subject

EXAMPLE Instructions on how to print the current *document* (3.1.19).

Note 1 to entry: In *printed information for users* (3.1.38), a topic is equivalent to a section (heading; subheading) and its content. In *onscreen information for users* (3.1.36), a topic consists of a title (heading) and information about a subject (typically, a task or a concept or *reference information* (3.1.43)).

Note 2 to entry: For onscreen information for users, the system sometimes presents a topic without *user* (3.1.54) intervention.

[SOURCE: IEC/IEEE 82079-1:2019, 3.43, modified — The EXAMPLE has been added; the original note 1 to entry has been replaced by notes 1 and 2 to entry.]

3.1.52
tutorial
instructional information (3.1.30) in which the *user* (3.1.54) exercises *software* (3.1.46) *functions* (3.1.23) using sample data that is supplied with the software or *information for users* (3.1.29)

3.1.53
usability
extent to which a system, product or service can be used by specified *users* (3.1.54) to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use

[SOURCE: ISO 9241-11:2018, 3.1.1, modified — Notes 1 and 2 to entry have been removed.]

3.1.54
user
person who performs one or more tasks with *software* (3.1.46) or member of a specific *audience* (3.1.7)

Note 1 to entry: Users can be, for instance, end users, administrators, computer operators, installers.

3.1.55
user interface
ensemble of *software* (3.1.46) and hardware that allows a *user* (3.1.54) to interact with a computer system

3.1.56
warning
advisory information in *information for users* (3.1.29) that states that performing some *action* (3.1.3) can lead to a potentially hazardous situation which, if not avoided, can result in death or serious injury

Note 1 to entry: See also *danger* (3.1.16) and *caution* (3.1.8).

3.1.57
window
area with visible boundaries that presents a view of a *software* (3.1.46) object or through which a *user* (3.1.54) conducts a dialog with a computer system

3.2 Abbreviated terms

CMS	content management system

DITA	Darwin Information Typing Architecture
FAQ	frequently asked question
GUI	graphical user interface
HTTPS	hypertext transfer protocol secure
JSON	JavaScript Object Notation
PDF	portable document format
SVG	scalable vector graphics
URL	uniform resource locator

4 Conformance

In this document, the word "shall" indicates requirements strictly to be followed in order to conform to this document and from which no deviation is permitted (“shall” equals “is required to”). The word "should" indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred, but not necessarily required (“should” equals “is recommended that”). The word "may" is used to indicate a course of action permissible within the limits of the standard (“may” equals “is permitted to”). The word "can" is used for statements of possibility and capability, whether material, physical, or causal (“can” equals “is able to”).

An organization may claim conformance to the requirements in this document for its information design and development processes, or for its information for users (products), or for both. Full conformance to processes is achieved by demonstrating that all of the requirements in Clauses 5 and 6 have been achieved. Full conformance of information products is achieved by demonstrating that all of the requirements of Clauses 7, 8, and 9 have been achieved.

This document is meant to be tailored so that only necessary and cost-effective requirements are applied to information for users. Tailoring may take the form of specifying approaches to conform to its requirements, or altering its recommendations and approaches to reflect the particular software and information product more explicitly.

NOTE The tailoring process is described in ISO/IEC/IEEE 12207:2017, Annex A.

This document may be included or referenced in contracts or similar agreements when the parties (called the acquirer and the supplier) agree that the supplier is required to deliver information for users in accordance with this document. It may also be adopted as an in-house standard by a project or organization that decides to produce information for users in accordance with this document. Tailoring decisions made by an acquirer should be specified in the agreement with the supplier.

When conformance to processes is claimed for a multi-supplier program, it may be the case that no individual organization 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 should record the tailoring of the required tasks, and their assignment to the various parties.

When conformance is claimed for information products, the organization or project should specify whether conformance applies to a single suite of information for users, a document set, or all information for users produced through the organization's information management and content management processes.

The fulfilment of specified requirements in this document does not release suppliers from their obligations to assess the legal framework for product compliance within each separate distribution area.

5 Information management process

5.1 General

This document covers the activities of information design and development as applicable for information for users of software.

NOTE 1 General requirements regarding the information management process are covered in IEC/IEEE 82079-1:2019, Clause 6.

In this context, design includes planning for design and development. Development should be performed concurrently with evaluation and test (validation and verification) of the information products. Design and development may be performed iteratively, particularly for information about software products that are being frequently updated or delivered in new areas.

Designers and developers of information for users work within the life cycle processes of the software product, which are defined in ISO/IEC/IEEE 12207:2017 and include the management of information for users within the information management process (ISO/IEC/IEEE 12207:2017, 6.3.6). The applicable requirements are the following: ”Define the content, formats, and structure of information items” (ISO/IEC/IEEE 12207:2017, 6.3.6.3.a.4) and “Obtain, develop, or transform the identified items of information” (ISO/IEC/IEEE 12207:2017, 6.3.6.3.b.1).

Thus, development of the information for users 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 information for users may be tested, distributed, and maintained together. The development of the information for users, including online and print information, should be a part of the development of the software product as a whole, not a separate activity. Although accurate information for users is not completed until the software has been fully developed, the information for users and the software product both benefit from concurrent development.

The classic development process of information for users applies to the life cycle of a single new software product. However, it is likely that software and information for users are designed and developed under more complex circumstances, such as the following:

— changes to software, systems, services, and other projects requiring the revision of existing content and the addition of new content;

— information-development projects independent of the development of new software, systems, and services requiring new and revised content;

— existing information for users to be converted to different formats or media, or provided in different languages;

— previous information for users adapted or used as models for different software products acquired or supplied by an organization.

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

Users of this document should adopt a style manual for use within their own organizations or adopt an industry-recognized style guide. Annex A provides guidance for the content of a style guide.

NOTE 2 ISO/IEC/IEEE 15289 provides recommended contents for the required documents throughout the systems/software life cycle.

NOTE 3 ISO/IEC/IEEE 26511 provides detailed information on information-development plans and the process of managing information-development projects.

5.2 Planning the information-development project

The development of an information-development project plan includes the following as defined in ISO/IEC/IEEE 26511:2018, 7.2.

a) Identify project goals.

b) Analyse project scope.

c) Take into account the target audiences.

d) Describe the topics to be developed.

e) Specify an information reuse strategy.

f) Describe accessibility requirements.

g) Describe usability requirements.

h) Describe translation and localization requirements.

i) Describe project deliverables.

j) Identify project tools requirements.

k) Identify quality, review, and testing requirements.

l) Determine the information-development project schedule.

m) Estimate the time and costs required to complete the project.

n) Analyse and mitigate risks.

o) Identify information-development project team members.

The information-development plan should be an integral part of the software development plan.

The information architect can, by way of assistance to the manager, estimate how much information is required, so that the estimate can be used in making choices about delivery methods and delivery media.

5.3 Managing the information-development project

Throughout the information-development project, the information project manager is responsible for the following activities so that the project quality and schedule are maintained:

a) Manage the project team.

b) Track project deliverables and schedule.

c) Manage project changes.

d) Maintain quality and the project vision.

e) Communicate with the software project team and management.

NOTE For more information about communication in an agile team and the information developer as part of the software development team, see ISO/IEC/IEEE 26515.

f) Manage reviews and tests.

g) Manage the translation process.

h) Manage final production and delivery.

6 Information architecture and development

6.1 Project requirements, objectives, and constraints

6.1.1 General

The job of managing the project requirements, objectives, and constraints is the province of the manager of the information-development project and it is described in ISO/IEC/IEEE 26511. However, in order to understand the requirements that affect the design for the components of the information for users, the information developer shall gather or receive information about the wider context of the whole project, as set out below:

a) the primary or intended purpose of the software product;

b) the primary or intended requirements and objectives for the software product (6.1.2);

c) the primary or intended requirements and constraints for the information for users, such as the policy, standard formats and styles for information for users set by the producer of the software product (6.1.3);

d) the intended acquirers or users of the software product and the information for users (6.2.2);

e) the protection of information that is required to be protected from being accessed by unauthorised parties;

f) the intended usability requirements.

If the stated requirements limit the design options so that users cannot be given a suitable set of information for users, the information developer 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.

6.1.2 Project objectives and goals

The information developer should expect that the software product or information for users will have to be updated at some stage, which can affect information design.

The information developer should be aware of plans for making modified versions of the software product available to users because modifications can affect the information design. Plans for software modifications should account for different levels of changes, including temporary fixes, interim versions, major upgrades or new releases. Plans should consider the scheduled and emergency change time frames and available methods for updating the information for users.

6.1.3 Requirements on information for users

The following information should be considered in constructing the information-development plan:

a) whether the organization has formal content management or information management policy and procedures;

b) what metadata is required for each piece of information for users to support content management, indexing, and searchability;

c) whether there are organizational standard formats, templates, styles, or systems; whether they are consistent with the requirements and recommendations in this document, and if they do not exist, whether they should be established consistent with this document;

d) international standards publications, such as those of ISO, IEC, and IEEE;

e) national standards;

f) industry standards for the system on which the product runs;

g) industry standards for the systems on which the onscreen information is viewed;

h) accessibility standards and requirements;

i) company, product, or operating system standards and conventions;

j) restrictions on copying or modifying the software and its information for users;

k) constraints on required content;

l) whether there is a need to provide background or conceptual information, such as the working procedures of the users' organizations or information about the software product itself, and if so, whether that information is readily available;

m) whether there are resources for supplying the information developer with technical information about the product, such as subject matter experts, written specifications, and the product itself (possibly development or prototype versions).

n) legal and other requirements, such as those relating to handling of private data.

6.1.4 Constraints on information for users

Information developers should be aware of legal and regulatory requirements. They should consider the following constraints and take legal advice if necessary:

a) requirements set by national legislation;

b) copyright status of the information itself;

c) copyright issues for content sourced from elsewhere;

d) acknowledgments;

e) trademarks;

f) any requirement that source code and information for users are to be kept in the custody of a third party until specified contractual conditions have been fulfilled (that is, escrow);

g) licensing;

h) any presentation requirements that apply (such as use of special logotypes on the packaging and in the information for users to identify proprietary platform-compatible information products);

i) intellectual property rights;

j) warranties and guarantees;

k) whether there is a previous version and, if so, which features have to be changed and which have to remain the same;

l) whether the product is stand-alone or part of a suite of products;

m) when the product can be available to the information developers (the product schedule can help determine the types and amount of information for users);

n) whether there are plans for future versions;

o) which platforms the product can run on at this release, and whether there plans for other platforms later;

p) whether the product is being developed for a specific organization or organizations, and if so, whether those organizations are the exclusive users;

q) whether translated, localized, or customized versions of the product are required;

r) data security.

NOTE One possible way for an organization to deal with clauses that cite “the information-development plan” is to specify that they are interpreted in the project plans for any particular information-development project.

6.1.5 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 information for users 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 information developer should consider whether these existing systems dictate the use of particular information delivery and viewing mechanisms.

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

a) using the software, so that information developers can find out how it operates and how users can use it;

b) writing and editing the text;

c) writing onscreen information and compiling help files;

d) drawing the illustrations;

e) reviewing the information for users;

f) testing the information for users;

g) conducting user tests, including tests of accessibility;

h) producing content for different media (audio, video, augmented and virtual reality, and others).

The information developers should be involved in designing, updating, and improving the formats and models for published output.

6.1.6 Schedule constraints

Information developers should, in addition to knowing their own schedules, also know something of the software product development schedule:

a) milestone dates;

b) whether there are work processes that need to be defined to change the way business is currently being done;

c) whether the software product is required to be submitted for certification and accreditation;

d) the major dependencies between different activities in the overall project.

6.1.7 Users and usability objectives

The usability of the information for users is an integral part of the usability of the software product and the information developers and software developers should work together to ensure usability. From the start, the primary intended user functionality and use should be considered and implemented into the structure, content, and format of information for users and supporting software; it cannot be added at the end of the project.

Therefore, information developers should be aware of usability requirements as stated in IEC/IEEE 82079-1. User needs, usability requirements and the method of confirming or verifying compliance shall be specified in the analysis and planning of information. Attributes included in usability include utility, comprehension, efficiency, retention, detection and reporting of errors, and customer satisfaction.

NOTE 1 ISO/IEC/IEEE 26513 and IEC/IEEE 82079-1 include recommendations for reviews and usability testing.

The goals to be measured in usability tests are the qualitative or quantitative targets. EXAMPLE 1 presents the process of defining usability testing with an example of video chatting with a group of people in an instant messenger application.

EXAMPLE 1

Step 1. Define the user's objectives.

The user's objectives are:

— to find the instructions for video chatting with a group in such formats and media as online help on desktop, mobile help, web-based help, FAQ, tutorials or 'Getting started' guide;

— to construct an e-mail; to define a group and add contacts;

— to start the video chat with the group and to end it.

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 navigation method used? Does the help text have to be reread 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 60 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 is performed correctly on the first try while following the instructions, then the help was acceptable, task-oriented, and complete.

Accessibility requirements extend usability requirements so that information for users is provided in media and formats that allow its use by those with vision, hearing, or other physical limitation.

EXAMPLE 2 The US government has published specific requirements for software accessibility, known as Section 508 of the Rehabilitation Act.

The process of developing information that it is suitable for an international audience, usually known as internationalization, and that recognizes national cultural requirements has a significant effect on the design of information for users, including presentation formats, writing style, and usability testing and therefore the information developer should determine whether the application can be used by an international audience.

For software products to be made available in other countries or regions using the same language as the source country, the information architect and developer should consider cultural issues, both in the software product and in the information for users, especially in examples.

NOTE 2 If the design is intended to be used with a content management system, see ISO/IEC/IEEE 26531.

6.2 Audience and task analysis

6.2.1 General

Along with understanding the requirements and constraints, an important activity before designing the information for users is audience and task analysis. This analysis determines who will be using the information and why they need to use the software.

The information architect shall perform these principal activities, using the audience (6.2.2) and task analysis (6.2.3):

a) determine what information shall be provided (the content) and what shall be excluded;

b) define navigation methods;

c) structure the content into necessary items in the document set, and determine the structure for each information product (6.3.1).

Subsequent clauses of this document address the structure and content required for the information for users. For guidance on deciding whether information should be delivered on the screen, on paper, via sound, or via some other mode, see 9.3.2.

6.2.2 Audience analysis

The information developer 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 tasks and use of the application are similar. The classification should consider:

a) the users' background, experience, and training;

b) the language familiar to the users;

c) the way that the application can be used;

d) the users' learning stages (e.g. how much experience they have with this application);

e) the frequency of use (some users use an application or feature frequently, some infrequently);

f) the users' working environments (do the users do most of their work at a desk, or in an environment that is noisy, hot, cold, or otherwise uncomfortable?).

NOTE An analysis of the tasks performed by members of each audience helps identify their information requirements (6.2.3). However, the above factors also 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.

For example, the following is a possible list of audiences for part of an order fulfilment system:

— Call centre operator

— Call centre supervisor

— Website manager

- Warehouse picking staff
- Warehouse manager
- Finance manager
- Accounts manager
- Accounts payable clerk
- Accounts receivable clerk

The information architect should use both a “bottom-up” and a “top-down” approach to check that all user types have been considered.

- Bottom-up: The information architect should consider who uses the software product and determine all the types of users. For some applications, the audiences may be identified by their job titles, such as accountant. For other applications, audiences should be identified by the role they are performing, such as data analyst.
- Top-down: The information architect 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 can be used to check the results of the bottom-up analysis.

User roles can correspond to people in a relationship that is not one-to-one; the same person can have more than one role: for example, salesperson and inventory taker.

Audiences can be grouped into a hierarchy, so that individual pieces of information for users can be targeted at several audiences. The information developer should use the hierarchy to group together audiences who have the same type of interaction with the software. The information developer should not (necessarily) reproduce the user organization's organization chart. Figure 1 is an example of such a hierarchy.

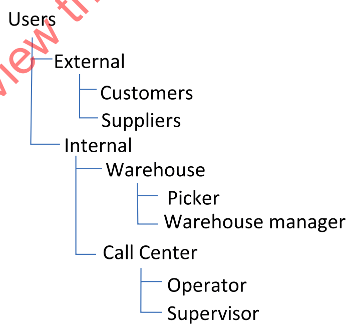


Figure 1 — Sample of an audience hierarchy

Audiences who are similar in other respects require different types of information for users 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 can be difficult to gather useful information about typical users' jobs and background, for example. In this case, the designer should concentrate on the tasks that the users carry out and the learning stages through which the user goes.

The information developer 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:

— Are printed documents carried around to different work sites?

— Are there convenient places to store documents?

— Do users have the equipment to view onscreen information?

— Do users have the equipment to hear the audio information?

If the information is expected 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 helps ensure that the material is presented in such a way that all users can find the information that they need, at the level of detail that they need. Layering usually involves presenting information in progressively more detail, using clear headings to enable users to select the right information. On a screen, layering may be achieved by various methods, including the use of hypertext links from overview-level items to items containing detailed information.

For each audience, the information developer should construct a persona that records the pertinent information. This persona is useful both in planning and as a guide for information developers and the software development teams. The persona may be recorded either in absolute terms (e.g. special qualifications of one audience) or in a comparison of users' experience to the norm. Table 1 illustrates the types of information that may be included in a persona.

Table 1 — Sample 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 can 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 attend 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 can be noisy and busy. Ticket agents sit at desks to work. Agents have access to a computer throughout the shift but can share printers. Offices have a ticket printer, scanner, and internet connectivity. Ticket agents use headsets for the telephone. Not all offices have: — shelf space to hold a small library of books; — drawer space in the desks; — wall space available for displaying wall charts.

6.2.3 Task analysis

When information content is structured into different topics or types of information products, the information designer should be aware of user tasks performed by various audiences.

The information architect 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 (e.g. if the software is still under development), then the information architect and developer should consider the tasks, or use other sources such as use-case design documents.

For example, the following is a sample task list for an instant messenger app:

- Send a text message.
- Send a photo, video, sticker or voice message.
- Start a video call.
- Check data security and compliances.
- React to a message with an emoji.
- Manage contacts.
- Manage groups.
- Turn notifications on or off.
- Change own status.

The information developer should consider the following information about each task:

a) why the task is carried out;

b) how frequently the task is carried out (to help determine whether users remember how to do it);

c) 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;

d) what discretion the user has in how or when the task is carried out;

e) in what sequence the user carries out the set of tasks;

f) what the prerequisites for the task are;

g) how fault-tolerant the task is; that is, how important it is that the user carries out the task correctly;

h) what the consequences of the task being completed or not completed are.

The information developer should group related or similar tasks, or tasks that involve similar steps, in a hierarchy as illustrated in Figure 2. The designer should record the details of each task in a task profile.

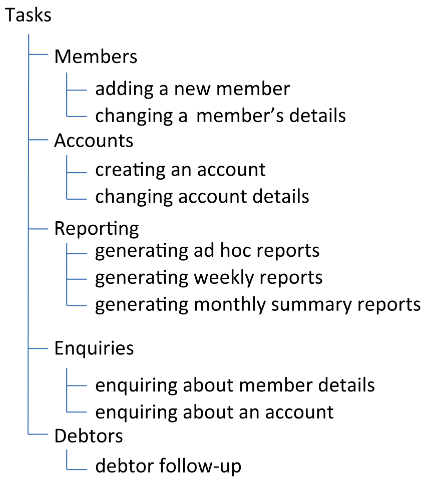

Tasks
Members
adding a new member
changing a member's details
Accounts
creating an account
changing account details
Reporting
generating ad hoc reports
generating weekly reports
generating monthly summary reports
Enquiries
enquiring about member details
enquiring about an account
Debtors
debtor follow-up

Figure 2 — Sample task hierarchy

The information developer should record which audiences carry out which tasks and should consider using a matrix, as illustrated in Table 2.

Table 2 shows, for example, which users carry out security-related tasks and what their learning stages are. As a result of this audience and task analysis, the information can 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 can be written for the task of viewing the security log to be included in both guides.

Table 2 — Audience mapping matrix

	Audience								
Task	**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 one's 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

Key to audiences

DIC	Data input clerk
SUP	Supervisor
MAN	Manager
AUD	Auditor
ACC	Accountant
FC	Financial controller
SO	Security officer
FA	Functional analyst
SA	System administrator

Key to learning stages

O	Occasional use
N	Normal use
E	Exploiting advanced features

6.3 Development

6.3.1 Information gathering

Information 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, surveys, questionnaires, instant messaging, e-mail, audio- or video-conferencing, Internet-based real-time chat tools, and other communication media used by the project team.

Information developers should first read the available software life cycle design documents, and then plan and conduct formal interviews with subject matter experts 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 can be perceived as interruptions by other team members. Information 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 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. Information 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 can be promptly corrected and any continuing information gaps can be identified.

NOTE For further information on design, information gathering and development, see IEC/IEEE 82079-1:2019, 6.3.2.

6.3.2 Developing information for users

The information developer should follow the guidelines specified in the information-development plan, using the specified tools and methods and prioritizing the information specified in the information-development plan.

The information designer and developer should have a consistent method and approach for applying changes to the information for users, similar to the process for developing new information products.

6.4 Review

Evaluation of information for users attempts to confirm that it is fit for purpose. Evaluations are performed throughout the information's development, production, and maintenance, from first draft to revision, for reuse or consideration of enhancements.

Tests validate and verify information for users in conjunction with the software. The information-development team should evaluate or test that the information fulfils the requirements at each stage of application development when the software is available, including unit testing and system integration testing, as well as usability testing, before the product is generally released.

The information designer and developer should have a consistent method and approach for applying changes to the information for users, similar to the process for developing new information products.

The following are typical checks to confirm that the content is of satisfactory quality.

a) Content is verified for accuracy by members of the software development team.

b) Content conforms to the style guide, and there are no grammar or spelling errors, or language in the text that can be difficult to translate.

c) Content is reviewed structurally so that each topic is stand-alone and answers only one question, topic structures conform to the original design intent (concept, task, reference, and troubleshooting), all content tagging is complete and accurate, and all content linking is consistent and logical.

d) Content is validated for usability.

   NOTE A detailed standard for the process of testing for usability and reviewing the information for users is found in ISO/IEC/IEEE 26513.

e) Customer feedback is obtained through usability testing and solicitation of comments through social media and other appropriate mechanisms.

f) The information for users is functionally correct.

6.5 Prototypes and drafts

Software prototypes may be used in developing information for users. Draft information for users may be developed in conjunction with software prototypes and often the information can be developed by reference to the functionality of software prototypes. In the case of embedded information, the prototypes also serve to validate that the mechanical aspects of the embedding process work correctly, that is, that the correct topic appears when required during the operation of the software. The correct topic is the one that relates to the functions that the user is using, and the one that is the correct localized version.

For each software prototype, the information developer should consider the following questions.

a) What is its purpose? What information is intended to be collected based on testing or trials of the prototype?

b) What is its scope? What part of the application is covered, and what types of information for users are included?

c) How are the tests and trials performed, recorded, and used in subsequent phases?

To avoid problems arising later in the project, the information developer should, in an early draft:

- try out any new methods and tools for creating and delivering information;
- produce a sample of the text and illustrations using the final production method;
- for embedded information, 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.

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

The information for users should not be drafted until the information-development plan has been authorized.

Further, the corresponding part of the software should exist, or (especially in an agile environment) 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 information developer should advise the project manager to review the project schedule.

Before the information 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 navigational features. The information developer should prepare a project glossary and apply it consistently. Clarity and simplicity of style are particularly important in documents that are translated. The information developer should avoid slang, jargon, and humour, and use similes and analogies with care.

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

- system design information;
- subject matter experts' or software developers' information;
- personal experiences of using the system or a prototype of it.

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

6.6 Content management during development

Approved versions of information for users shall be maintained securely and separately from copies checked out for development. Each draft issued for review shall be uniquely identified. Backup copies of information for users should be stored securely and separately from the systems being used for information development.

6.7 Translation considerations in design and development

If translation is required, developers should not embed text as a bitmap in graphics or in videos. Instead, they should use some graphic format that permits the text to be held separately from the image, such as scalable vector graphics (SVG) format, or some proprietary graphic formats.

If text is rasterized, the original application files for illustrations should be retained so that text can be edited or translated and new rasterized illustrations recreated.

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

NOTE 1 Annex B provides a style guide for translation.

Different languages require different amounts of physical space to express the same thing. Developers of information for users should be aware of this, for example, when designing graphics, since a call-out caption in Japanese, for example, can require more space when translated into English, and more still in Finnish.

NOTE 2 It is possible that a target language does not have a term that corresponds to a term in the source language. In these cases, the translated text is often many times longer than the source text.

Translations should be made by native speakers of the target language, not by native speakers of the source language, and 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 information for users shall be translated only after the translated list of terms has been approved.

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

NOTE 3 Translation services and localization are discussed in ISO 17100.

6.8 Final assembly and review

When the final draft has been approved, publishing/production staff, editorial staff, or a different person than the one who wrote the information for users, shall:

a) proofread the text; although automatic spelling and grammar checkers and other authoring tools can be helpful, make visual checks;

b) carry out a final check of cross-references and hyperlinks;

c) if non-print media are part of the information for users, check that they are present and functioning;

d) check that the illustrations are placed correctly (9.14) and are clear enough for the chosen method of presentation;

e) if the information for users is to be printed, review final pagination to ensure that lists or short sections are not split over page breaks and that headings are not separated from the following text by page breaks;

f) in the case of information that is embedded in the software, check that the information appears correctly to the user as required and requested;

g) check that all the pages or topics are present;

h) check that all the headings, tables, and figures are correctly numbered.

6.9 Packaging

The developer should at least consider the following:

a) whether the software product and the information for users are to be packaged together;

b) whether there is a house style for packaging;

c) what methods are used to deliver the product and information for users.

6.10 Release

The activities required to release the information for users depend on the degree to which it is integrated with the software, which depends on the medium in which the information is delivered. Most information for users falls into one of the broad categories:

- onscreen material that are separate from the product, such as a website;
- onscreen material downloaded from a website;
- printed or printable, including PDF format;
- embedded in the product;
- video;
- sound.

6.11 Updating and maintenance

Properly designed and developed information for users is more easily updated and maintained. The information architect should expect that modifications are likely, and can result from:

- the creation of a new, customized, rehosted, or upgraded version of the software product;
- the discovery of errors in the existing software product or information for users;
- the development of new software products for which information for users can reuse existing content.

6.12 Version control and change control

A configuration management (CM) process shall be used to control the information for users. The designer of the information for users should consider the CM policies that may influence the information architecture and affect its maintainability.

NOTE 1 Further information on CM can be found in ISO/IEC/IEEE 26531.

Change control procedures for the project shall take account of the requirements for information development. Information developers should be members of any body responsible for approving changes. Information for users can 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, information developers should be informed promptly, so that they work only with up-to-date information.

Small changes to a system can cause major changes to information for users, while major changes to a system do not always require major changes to the information for users. For example, altering a system menu can be a small change to the software itself but can 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 can be a major task in developing the software, but not affect the user's view of the software at all, and therefore can require no changes to the information for users.

The same CM system used for the software development may support the information for users. It usually has some features appropriate for the information for users.

NOTE 2 IEEE 828 provides guidance for configuration management processes.

7 Information quality

7.1 General

As required in IEC/IEEE 82079-1:2019, 5.3.1, information for users shall be correct, consistent, comprehensible, concise, minimal but complete, and accessible.

7.2 Correctness

Correctness is the degree to which a system or component is free from fault in its specification, design, and implementation.

Information for users shall correctly reflect the actions and expected results of the applicable software product version.

NOTE Correctness of information can be verified following the requirements in ISO/IEC/IEEE 26513.

7.3 Consistency

The structure and layout of information for users shall be consistent. Similar information, such as concepts, tasks, and references shall be formatted in the same manner throughout the information for users.

Consistency of layout applies to such items as screens, pages, text, headings, blank spaces, vertical spacing, graphics, icons, colours, typefaces, signal words, lighting in videos, backgrounds, caption style, and acoustic signals. The information developer should provide guidance to software developers to apply consistent and correct terminology in the user interface and messages so that the information for users can be consistent with the interface.

Information for users shall use consistent terminology throughout an information set and for elements of the user interface, data elements, field names, tasks, pages, topics, procedures, and processes.

7.4 Comprehensibility

Where there is a wide variety of users, with a wide variety of experience, skills, and knowledge, the information for users should be comprehensible (adequate for use) by the least experienced of the expected users.

Terminology that is common in the user's environment or the application area should be used to aid comprehension. Terminology that is common in the software developers' environment, but not in the user's environment, should be avoided.

NOTE Comprehensibility of information can be validated by usability testing as detailed in ISO/IEC/IEEE 26513.

7.5 Conciseness

Information for users shall be concise with respect to format and media. Although educational material may use repetition to promote retention, information for users should avoid duplication of unnecessary details.

7.6 Minimalism

Minimalism is an action- and task-oriented approach to instruction and information for users that emphasizes the importance of realistic activities and experiences for effective learning and information seeking.

Minimalism shall be applied to all information for users. Minimalism is a user-centred design approach. The main purpose of minimalist information and training is to facilitate task performances (e.g. software use). Minimalism is user-centred because it considers audience characteristics such as prior knowledge and context of use in designing instructions.

In a minimalist approach, the goal is to provide only that information that the users need to complete a task (or for the users to reach their goals).

Minimalist information is often contrasted to approaches that aim for full coverage of all possible product functions. Such excessive documentation tends to be wasteful and dysfunctional. Too much product information can hinder users from comprehending how to use a product and obstruct quick access to the pertinent information the users need to achieve tasks.

7.7 Accessibility

Information for users shall be accessible to and usable by the expected groups of users in their environments. It shall be technically available, legible, and findable by the target audiences in the context of use and throughout the declared service life of the supported software.

Working with the information-development project manager, information architect, or team leader, the information developer shall assist in specifying the accessibility requirements for an information-development project. The accessibility requirements may be developed with assistance of user-experience experts in the organization.

Information that is distributed through a website or mobile application should meet the accessibility guidelines given in sources such as:

— IEC 40500;

— ISO 9241-20;

— ISO 9241-171;

— US Rehabilitation Act, Section 508.

NOTE For typical requirements for information accessibility, refer to ISO/IEC/IEEE 26513:2017, 7.4.

8 Structure of information for users

8.1 Overall structure

As required in IEC/IEEE 82079-1:2019, 8.1, information for users shall be structured so that the content is easy to find, use, and understand. The structure shall promote ease of access to the user's desired content and consistency in presentation. Information for users shall be consistently structured, depending on the information type.

8.2 Modularity

Modular information (presented in topics which are self-contained) is both easier to manage and easier to find, understand, and apply. Modularity should be achieved by applying these principles:

a) Content is developed as logical, standardized, and standalone topics.

b) Topics are independent of previous modules and are self-contained.

c) Modules are standardized based on both the type of content and the users.

d) Designing self-contained topics forces the information developer to be concise and focused.

e) Topics are labelled with clear and meaningful titles.

f) Topics should be complete, in that they fully cover what their headings promise to cover.

g) Related topics are linked.

8.3 Structuring by function

Information for users shall be structured according to the functions that users perform, including understanding concepts, performing tasks, accessing reference data, and troubleshooting problems. Information types include the following.

a) Conceptual information, which provides users with background information about a product, interface, or task before using a function to perform a task. Conceptual information may explain a product, the purpose of the product and its features, and its relationship to other products and may help users orient their knowledge of a software product to the tasks they wish to perform. Conceptual information generally answers the question: "What is …?"

b) Instructional information, which provides users with information for performing a task. Instructional information generally answers the question: "How do I …?"

c) Reference information, which provides factual information about software to support task performance. Software reference information may also include error codes, keyboard shortcuts, language elements, commands, functions, and API information.

d) Troubleshooting information, which provides methods for isolating software or system problems and resolving them.

e) Glossary of terms.

Information for users shall be developed as a series of topics, each assigned to an information type, so that each topic is developed consistently. Each topic should be understandable in itself, out of context.

Topics shall be organized into sequences, hierarchies, or groups that enable users to locate the information they need quickly and easily. Topics shall include metadata that identifies the intended audience, the product, the information type, and other data that enable the users to identify appropriate information to understand and successfully complete a task.

NOTE Detailed information on developing a metadata schema is available in ISO/IEC/IEEE 26531.

Lengthy information for users should be presented in sections that correspond to functions that users are most likely to perform. For example, information for users may be organized according to tasks performed in starting a software application, performing simple to complex tasks, and correcting problems. Such information may be divided into larger logical units, referred to as chapters in print or PDF and as screens or topics in online information.

As stated in IEC/IEEE 82079-1:2019, 9.3, advertising or promotional content shall be kept separate from the information for users and viewing of such content shall not be required prior to viewing the information for users.

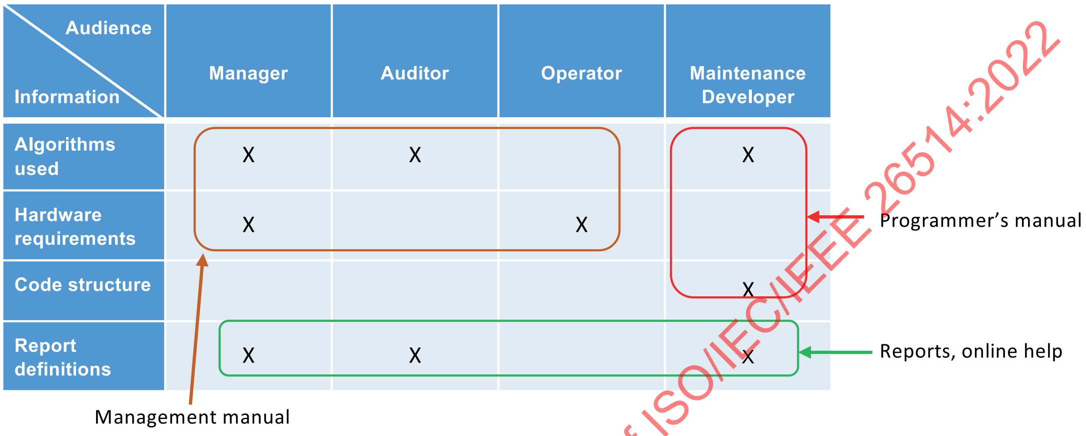
8.4 Information model

An information model represents concepts, relationships and rules for information and helps to structure information. An information model shall be applied to individual topics, an information set for a single system, and to a single information product.

Information models may be developed by the information-development organization or adopted from an open source or standard industry information model.

An information model shall reflect the needs of the intended users, specifying the information needs of a range of users, including new and experienced users. Users with different organization roles, including managers, developers, and operators, should receive information that is targeted to their information requirements.

Figures 3 and 4 show how the users' information needs can be supported by the structure of the information set.

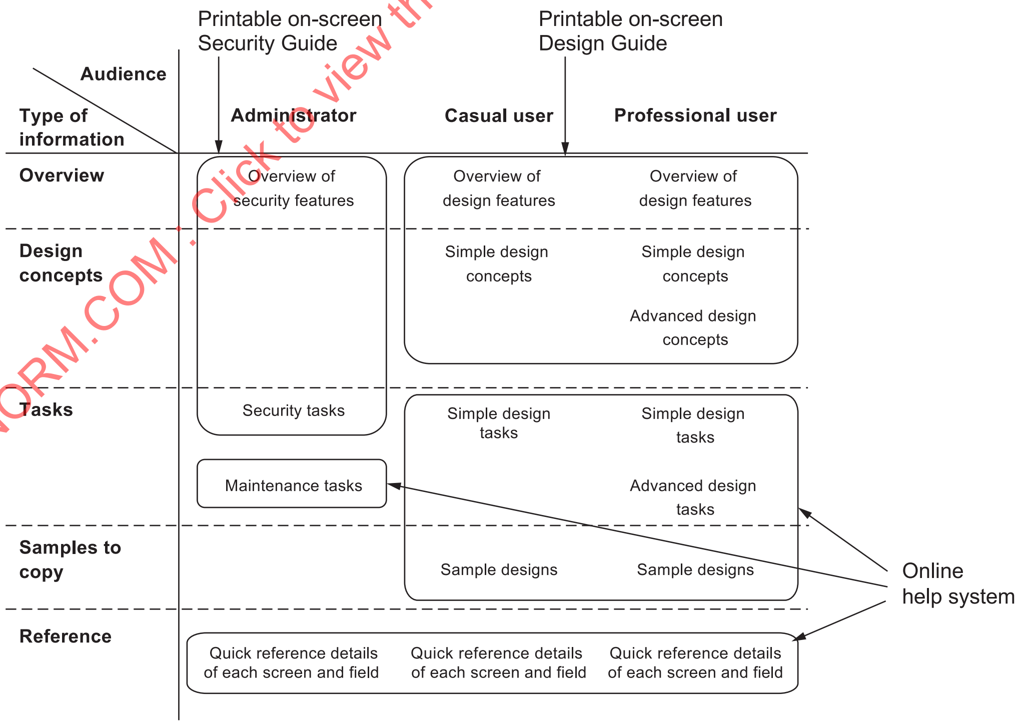


Information \ Audience	Manager	Auditor	Operator	Maintenance Developer
Algorithms used	X	X		X
Hardware requirements	X		X	
Code structure				X
Report definitions	X	X		X

Key

X this audience needs this information

Figure 3 — Using users' information needs to determine information content



Type of information \ Audience	Administrator	Casual user	Professional user
Overview	Overview of security features	Overview of design features	Overview of design features
Design concepts		Simple design concepts	Simple design concepts Advanced design concepts
Tasks	Security tasks Maintenance tasks	Simple design tasks	Simple design tasks Advanced design tasks
Samples to copy		Sample designs	Sample designs
Reference	Quick reference details of each screen and field	Quick reference details of each screen and field	Quick reference details of each screen and field

Printable on-screen Security Guide
Printable on-screen Design Guide
Online help system

Figure 4 — Using users' information needs to determine information delivery methods

Organizations shall require that all information developed conforms to the defined structures in the information model.

NOTE Additional information on specifying an information model is available in ISO/IEC/IEEE 26531.

8.5 Structure of conceptual information

Conceptual information provides users with information they need to reach their goals and complete tasks successfully. Each concept topic shall include these structural elements:

a) a concept title;

b) a concept body, consisting of sections, definitions, paragraphs, illustrations, and examples;

c) cross-references to tasks or other relevant information.

The concept title shall consist of a noun phrase identifying the subject matter of the concept. For example, a concept supporting users of conference call software may be titled “About Conference Calls”. Information developers shall develop a naming convention for all concepts so that they are named consistently through the information for users.

The concept body can begin with a short description of the concept, followed by a series of sections, definitions, paragraphs, illustrations, and examples to explain the concept to the user. Subsections, examples, and definitions can be introduced with subtitles in the body of the concept.

Cross-references provided in concept topic should point the users to related concepts, tasks, or reference information.

Concepts should be restricted to information required by the users to successfully complete tasks. Information developers should avoid adding conceptual information that can be interesting, but not necessary for task completion.

8.6 Structure of instructional information

Instructional information provides users with the information they need to complete tasks successfully. The requirements of IEC/IEEE 82079-1:2019, 8.3.4 shall apply. Each task shall include these structural elements:

a) a task title;

b) preliminary information, consisting of the purpose of the procedure, explanations of concepts, or cross-references to the explanations, identification of prerequisite activities, list of resources needed, and relevant warnings and cautions applicable to the entire procedure;

c) instructional steps, and possibly sub-steps, examples and troubleshooting actions;

d) completion information, including references or links to related topics and how to determine if the procedure has or has not completed successfully, and possibly troubleshooting information.

NOTE 1 The preliminary information, instructional steps and completion information can be referred to as the task body.

The task title shall consist of a verb phrase identifying the subject matter of the task. For example, a task supporting users of conference call software can be titled, “Transferring a call.” Information developers shall develop a naming convention for all tasks so that they are named consistently throughout the information for users.

The task body should begin with a short description or overview of the task, if that short description provides a useful context of the task for the users.

Each task step shall be numbered. The task steps shall be presented in the order of performance. A task step shall consist of an imperative phrase if the imperative exists and is culturally acceptable.

EXAMPLE A conference call task begins with the step: “Dial the conference call number.”

Task steps may be supported by additional information to assist the users in performing the step correctly. Additional information includes warnings and examples.

Steps may include troubleshooting information in those cases in which a user is more likely to make a mistake. Immediate troubleshooting information assists the user in successfully completing a task without having to seek separate troubleshooting information.

Each task step shall consist of one action only. The next action shall be written as a new step and not combined with an existing step.

Each task step may include a series of sub-steps, structured in the same manner as the primary task steps.

If the user has several options for completing a step, the choices may be presented as a table or flowchart, with if/then headings or other choice points.

Each task shall be closed with completion information, which is a task result that explains what should have occurred if the task has been completed correctly. Each task may include examples of the completed task, such as a code example or an image of a completed data-entry screen. Each task may specify a post-requirement, alerting the user to actions that are needed after the task is complete.

Tasks with a long series of steps should be structured into a series of smaller tasks with an overall process description to show how the smaller tasks are related to each other.

Task-oriented information may include tutorials to assist new users in understanding task completion. Examples in the tutorials should be tested to validate that they can be easily and properly performed.

NOTE 2 Structured authoring is defined by the OASIS Darwin Information Typing Architecture specification. The structure of topics is defined in section 2.7 of that specification.

8.7 Structure of reference information

Reference information supports task performance by providing details that the user needs to perform tasks successfully. Software reference information includes command reference lists, specifications, error codes, parameters, glossaries (8.11), and more. Each reference item shall include these structural elements:

a) a reference title;

b) a reference body, consisting of lists, tables, or other information resources;

c) cross-references to concepts, tasks, troubleshooting, or other relevant information.

A reference title shall facilitate ease of access and clearly communicate the substance of the information. For example, a reference title for a conference-call application can provide a list of options for notifying call members of the date and time of the call. Information developers shall develop a naming convention for all reference topics so that they are named consistently throughout the information for users.

For context-sensitive help topics or popup-help topics, titles or headings can be omitted.

References shall be structured so that like information is presented using the same structure. For example, if a reference consists of a command dictionary, each entry in the dictionary shall have the same structure. If a reference informs users about terms, values, or descriptions, each such reference shall have the same structure. If tables are used to list and identify items, tables through the information for users shall be structured in the same manner, including column headings and values presented in each column.

References shall be organized for ease and speed of access. A list of commands or error messages should be arranged in lexical order, alphabetically, or by message number.

8.8 Structure of commands

The information developer shall explain the formats and procedures for user-entered commands, including syntax, required parameters, optional parameters, default options, and order of commands. Information may be provided on the development and maintenance of macros and scripts.

Examples should illustrate the use of commands. To find a solution to a problem, users usually search for concrete examples, similar to their own problem situation.

Information for users shall explain how to interrupt and undo an operation during execution of a command and how to restart it, if possible. Information for users shall describe how to recognize that the command has successfully executed or abnormally terminated.

8.9 Structure of troubleshooting information

Troubleshooting information provides users with methods for identifying and solving problems. Each troubleshooting topic shall include these structural elements:

a) a troubleshooting title;

b) a troubleshooting body, consisting of steps to diagnose and resolve a problem;

c) cross-references to concepts, tasks, other troubleshooting, or other relevant information.

A tree diagram or flow chart is often useful in diagnosing errors and their causes, with recommended corrective actions. A troubleshooting title shall clearly identify the problem, using words that the user is likely to use in describing the problem.

8.10 Structure of error messages

Software systems should provide messages to users to confirm normal operation, especially when transaction processing does not return immediate results, and to notify the users of errors. Reference information shall include each error message with an identification of the problem, probable cause, and corrective actions that the user should take.

As early as possible, information developers should provide guidance to software developers in regard to wording of error messages so that they are clear, concise, accurate, and useful.

The information developer should avoid the following when constructing messages:

a) technical jargon and system-oriented information (except in messages for technical staff);

b) abbreviated forms of words or terms and contractions, which can slow comprehension, especially in technical messages, and be difficult to translate successfully;

c) phrases that blame the user or imply user error.

8.11 Glossary of terms

Information for users shall include a glossary, if terms or their specific uses in the user interface or information are likely to be unfamiliar to novice users in the audience. The glossary shall include a lexically ordered or an alphabetical list of terms and definitions. The glossary may also include information items such as field names including their definitions and manner of use in the interface. The glossary should define vocabulary in user's terms, not in software developer's terms. The terminology and the order of terms in a glossary can be changed by translation.

8.12 Structure of printed information for users

Information for users can be developed as print publications or delivered through websites as individual topics or delivered as complete publications that emulate print publications, or a combination. As required in IEC/IEEE 82079-1:2019, 8.4.2, print publications shall be structured for ease of access and should include the following sections:

a) title page (required);

b) copyright information, including document version and date published, and software product and version (required);

c) table of contents (required for information exceeding 12 pages);

d) introduction (optional);

e) topics, including concepts, tasks, references, and troubleshooting (required);

f) index (required for lengthy and complex information for users).

Information for users comprising more than two printed pages shall be numbered. Tables of contents shall have the same numbering as the pages in the text and sufficient levels of headings to help the users find information efficiently. No more than four levels of headings shall be used. Introductions shall focus on information needed by users to understand the purpose and scope of the document. Introductions should not include advertising information.

The index shall contain terms most likely to assist users in finding information quickly. The index should not consist solely of the titles of the topics, even those revised to appear to be valid index entries.

8.13 Structure of online information for users

Online information for users may be delivered as a PDF. However, other delivery methods (e.g. video and sound) are available that can make information more accessible. If information for users is delivered as a PDF, the table of contents and the index should enable users to move interactively and directly to the relevant information.

Information for users should also be delivered as individual topics based upon user queries. Individual topics should include references to related content and to content that appears before and after the selected topic. A table of contents should also be displayed. See IEC/IEEE 82079-1:2019, 8.4.3.4.

As required in IEC/IEEE 82079-1, a mechanism for electronic search shall be provided for information for users that is delivered electronically (9.10.7).

8.14 User-generated content

8.14.1 General

User-generated content has become increasingly important to organizations that want to take advantage of the knowledge that users have about their products, particularly knowledge accumulated from actual implementations. User-generated content can benefit the entire customer base, as well as the internal product and information developers. As a result, onscreen information for users may include facilities to enable users to add their own content or customize the existing content.

The facilities provided could include adding company logos, lists of permitted values, simple annotations to existing topics, or links to the user's own onscreen information systems, documents or videos (9.13).

User-supplied or customized content may be:

— global, which can be seen by all users;

— group, which can be seen by specific groups of users;

- local, which can be seen only by the user supplying it.

However, sustaining user-generated content can be difficult and complex, because:

a) many users are reluctant to devote time and energy to contributing content to the organizations from whom they purchase products;

b) users are already contributing content in places that are not readily monitored by the internal information developers;

c) user content does not provide the best methods for solving a problem or using the product, or can contain errors;

d) information developers lack scheduled time to monitor user contributions;

e) information developers do not know what methods or avenues are best to encourage high-quality content from customers.

f) the user-generated content can be incorrect when the software is updated.

The acquirer should consider what will happen to the user-modified information when the application or standard information for users is updated, since it is not known whether this customized information will still apply to the new version of the software product. The supplier may provide an automatic method for incorporating this information into the new version of the supplied onscreen information for users.

8.14.2 Goals and practices

Organizations and information developers shall establish clear goals for the implementation of a strategy for user-generated content.

Possible goals include the following:

- build user loyalty and satisfaction by engaging users in prioritizing technical information;
- reduce user support costs and increase the speed and accuracy of information;
- have users set requirements and priorities for new products, features, and company-generated content;
- improve content by incorporating user comments and suggestions, which can be extracted and prioritized using AI or machine learning;
- supplement company-generated content for highly configurable products with a wide variety of customer-generated examples, scenarios, and best practices;
- incorporate user-generated content to support specialized content such as APIs, commands, and system integrations that have detailed and specific information challenges;
- provide more interactive and timely troubleshooting for users;
- answer users' questions about using the product more quickly and accurately for new products and those that have a niche market.

If a program to promote user-generated content is implemented, organizations should adopt a series of practices:

a) dedicated SME resources to contribute initial content and provide support for difficult questions;

b) incentives for customer contributions;

c) a carefully defined model for contributed content;

d) information-development resources to monitor, edit, and incorporate content into company-generated content;

e) easy-to-use tools for authoring;

f) easily accessible sites for hosting information;

g) tools to provide analytics on use, search, and contribution;

h) a set of defined steps to preserve intellectual property rights and to clarify what liability the supplier and the customer have for customized information for users.

To help a user-generated content initiative, organizations should be aware of common pitfalls and potential mistakes.

— Time and resource commitment: Soliciting, monitoring, tracking, and editing user-generated content can be extremely time-consuming, especially when done manually. Without adequate tools to handle workflow and integration, information-development teams can struggle to sustain the workload.

— Lack of focus and follow-through: User-generated content requires commitment and regular participation from all involved. Without dedicated resources and continued involvement of the sponsoring company, the community can decline.

— Poaching of employees: Some customers want to protect the identities of their employee experts who contribute in a public or semi-public forum.

— Companies that discourage their employees from spending time contributing content. Investigate customer policies about employee contributions through surveys or interviews. Provide incentives to the customer companies to encourage participation.

If information for users is supplied in modifiable form, the supplier shall manage modifications so that:

— the user cannot delete information which is important for others to see, such as critical warnings;

— mistakes can be corrected;

— the original information can be restored as supplied.

The format of a wiki is generally suitable for user generated content, as it allows comments and discussions on the manufacturer's information for users.

For open-source software there should be expanded facilities for user-generated information for users, such as wikis and links to user forums or to help and service websites. A guideline about how to write helpful information for users may be provided to the open-source community for the purpose of standardization and quality control.

8.15 Application programming interfaces (API)

8.15.1 General

An API is an interface to an external system or service, often running on a remote computer and written by a separate developer or developers. The external system provides a library of functions which can use protocols (e.g. HTTP) and objects of different formats (e.g. JSON), and have defined parameters. Software developers can use these functions and knowledge of the protocols, parameters and objects to create software that interacts with the external system or service.

APIs can be integrated into a software development kit (SDK).

The external system usually implements a library of methods and classes, presented to the user as a library of endpoints, an endpoint being the point at the end of a communication channel at which an API connects with the external software resource.

NOTE The term 'API' is sometimes used colloquially to refer to the endpoints but 'API' as used in this document is the interface to the endpoints.

The endpoints can be invoked in a manner similar to the way subroutines with lists of parameters are invoked in traditional programming languages. As a result, it is not necessary to know the internal operation of the external system or details of the algorithms implemented, or the programming language or languages used in the external system. The API hides the nature of the underlying system.

APIs can, for example, allow developers access to applications which can manage, for example:

- graphics;
- file systems;
- networking;
- factory automation;
- instant messaging;
- security functions.

It is useful to distinguish at least two types of API:

- APIs that resemble subroutine calls;
- web APIs.

EXAMPLE 1 An API that resembles subroutine calls. This connects to an endpoint that returns all information about users identified as UserID and belonging to a specific OrganizationalUnit in an organization:

```
Get_DigitalIdentity (UserID, OrganizationalUnit)
```

EXAMPLE 2 A web API. This connects, via an HTTPS call, to an endpoint that returns a list of channel activity events that match the request criteria

```
https://www.googleapis.com/youtube/v3/activities
```

8.15.2 Providing information about an API

8.15.2.1 General

Information for users of an API should include one or more of the following: a 'getting started' or quick start guide, a developer guide, and an API reference.

8.15.2.2 'Getting started' or quick start guide

A 'getting started' or quick start guide should:

a) describe the API;

b) explain the purpose of the API;

c) describe the main endpoints of the API;

d) explain the structure of the API and typical workflow.

8.15.2.3 Developer guide

The developer guide explains how to use an API while developing software. For each endpoint in the API, it explains the operation of the endpoint and the parameters.

A developer guide should contain:

a) system prerequisites necessary for deploying the API libraries;

b) the list of packages of the API library with a short description of the functional area covered by the package;

c) the main workflows, made up by a list of API calls suitable to get a specific result;

d) the list of the main data format managed by the API.

8.15.2.4 API reference

The API reference shall contain detailed information about the API endpoints including parameters, data types, response codes, the nature and dynamic of the responses, error codes, exception conditions, and examples of input and output data. It shall contain all information needed for users of the API, usually programmers, to use it properly.

If this information is auto-generated from source code or otherwise, information developers should provide support, review, and editing services.

8.15.3 Structure of the API reference

8.15.3.1 General

All endpoints of the API shall be described and the descriptions shall contain, for each endpoint:

a) name and structure;

b) a brief description of the action that can be performed using it;

c) name, data type, sequence, and set of values of each parameter;

d) payload information;

e) request and response information;

f) error codes.

The descriptions may also contain the following:

— request examples;

— response examples.

These sets of examples should be consistent with each other.

Any code given as an example should, if practical, be suitable for copying and use in the API.

8.15.3.2 Parameters

Every parameter shall be described. The description shall contain at least the following:

a) data type: examples are integer, character string, date;

b) description: a short text to explain the role of the parameter and possible constraints to be considered;

c) permitted values: parameter values can be constrained in a number of ways, such as assuming a specific set of values, a range of values, or a value to be validated in a look-up table.

Table 3 shows an example of a parameter description.

Table 3 — Example parameter descriptions

Parameter	Data type	Description	Required	Data values
Timestamp	String	Contains the timestamp indicating day and time associated with the data package production.	YES	The format of the timestamp conforms to the ISO 8601 series.
alertThreshold	Integer	Integer value indicating the alert value that triggers an alarm.	NO	0 through 10. If the value is 0, the alarm will never be triggered.

8.15.3.3 API payload information

API payload examples should be provided. API calls are usually made with a payload structured according to a specific format, for example, JSON, and if so, the example should be formatted in the manner customary for that format.

EXAMPLE A payload in JSON format:

```
{
"templateId": <templateId>,
"description": “<description>”,
 "name": "<name>",
"enabled": <true/false>
}
```

8.15.3.4 Request and response information

All response codes that can be returned shall be listed. Table 4 shows an example.

Table 4 — Sample HTTP response codes

HTTP status	Description
200	OK
400	Bad request
404	Not found

8.16 Frequently asked questions (FAQ)

Frequently asked questions (FAQ) can supplement well-structured instructional, reference, and troubleshooting information or a minimal quick reference guide. They can provide a stopgap for situations when problems and issues are reported. FAQ may be used to communicate temporary situations, such as workarounds for newly reported issues. As soon as feasible, the information in the FAQ should be archived or incorporated into the topics of the information for users.

To create useful FAQs, information developers should research user questions by defining use cases, performing usability tests, tracking user search input, and evaluating user questions on customer service. FAQs can allow users to provide feedback on existing questions or to submit new questions.

The answers to FAQs can contain step-by-step instructions with screenshots or tutorials. It should be possible to link from one question to another and also to link to the relevant topic in the information for users. The answers to questions shall be compact and contain only necessary information (following the principle of minimalism). The questions should be current to the published software versions and not outdated. The FAQs should be updated with their most recent date of publishing and the applicable software versions, so that the user can trust that the FAQs are relevant.

FAQs should be accessed from the help menu of the software, integrated into the online help itself as separate topics, or published in the support area of the software provider’s website. The FAQs should include a search function.

If there are ten or more questions, the questions should be subdivided into thematic categories. Examples of categories can be installation, login, customizing, managing, and so on.

Categories should not be developed for different user levels such as beginners or advanced users because they are difficult to differentiate, especially for FAQs. The questions should be ordered in a logical way, such as by importance, frequency, or relevancy. The questions should be formulated consistently, e.g. in question form or through the use of common word structures. The FAQs can be displayed in expandable and collapsible formats.

8.17 Chatbots and voice response systems

Chatbots and voice response systems (VRS) use artificial intelligence (AI) software to interpret key words in users' queries and provide related information.

Design for chatbots represents a transition from the design of visual layout and interaction mechanisms to the design of conversation. Instead of explaining to the user which content and features are available and which steps to take to reach the desired goal, the user interaction has to be interpreted to understand what the user needs. To make the interactive content really helpful for the user, methods of self-learning (deep-learning) systems and artificial intelligence (AI) should be applied.

The chat tool needs to be able to handle common misspellings and alternate wordings (e.g. in English, letter O for zero). Many chat tools can also send images and emojis, even videos as well as words.

At their best, these methods can save time for users, compared to searching through a website or mass of information for users. They also save time for human customer service agents responding to common service requests. However, they can be frustrating if poorly designed and built with repetitive requests for the same information, looping functions or dead ends, little information beyond what is already shown, inability to acknowledge that the user's question is about a function the software doesn't have, or no recourse to a human who can deal with complex or unusual questions.

NOTE This subclause applies to VRS and chatbots that provide "how do I" task-oriented or reference information. In highly developed form, VRS become personal assistant systems capable of machine learning to answer a wide range of queries and control other systems. These utilities can also be used to accomplish application transactions or to qualify prospects for sales (out of scope for this document).

In planning for a chatbot or VRS system, the designer should identify:

a) the topics about which information will be included;

b) alternate words which the user may use to ask about a topic;

c) what information is needed from the user to answer the question (e.g. product name and model number, account number, or version number);

d) the minimum number of branches or questions needed to get the user to the correct information: ideally, no more than three, or perhaps up to five;

e) short answers to each question, remembering that users may be accessing the chat on a small-screen device.

f) backup answers in case the user does not provide needed information or decides to leave the chat;

g) appropriate apologetic responses and referrals when the chatbot doesn't understand the question or doesn't have the information.

The possible dialogs need to be completely mapped out (e.g. through a flowchart) so that there are no dead ends.

System dialogs need both openings and closings. In the opening, the chatbot should greet the user and offer to help. Sometimes chatbots are personalized and give human-like introductions (as, for example, "Hello, I'm Chris, how can I help?"). Bots can also express approval or apologize in context; however, it's important to be clear whether the user is interacting with a human or a bot.

Next, the chatbot repeats what the user requested, or paraphrases it in the terms used in the system for confirmation that the question has been properly interpreted. If confirmed, the dialog can then proceed to obtain more information from the user and provide a useful response. The chat can also present two or three alternatives if the first question could lead to several topics.

For the closing, the dialog should confirm that the user's request has been answered, and then ask if there is anything else needed, in case the user needs to ask additional questions. It should be clear from the user's response that they are ready to end the dialog. The chat should end with thanks and encouragement to return.

With natural-language interfaces and chatbots, stronger attention to ethics and privacy is needed. Interactive content facilitates a better interaction between the content creator or the support staff and the audience. As the attention span of the audience is extremely limited, the interactive content should be efficient to use in terms of time and effort. Acceptance tests can be a proper method to evaluate the efficacy of interactive content.

9 Format of information for users

9.1 General

This clause covers aspects of the presentation of information for users:

- how to present information so that it is easily understood;
- what facilities to provide for navigating the onscreen information;
- what kind of media and format to use.

Information for users that is embedded in software shall be easy to distinguish from the software functions.

9.2 Consistent format

As required in IEC/IEEE 82079-1:2019, 5.3.6, information for users shall be consistently formatted, depending on the information type.

Users can use information more easily if it is consistent. As part of its content strategy, the organization shall set conventions and provide style guides for the whole product, project, or organization, typically for entire documents and document sets, topics, chapters, front matter, back matter, prefaces, terminology, icons and graphical symbols, navigational controls, and system messages. The organization shall provide style sheets and templates to support information developers in maintaining consistency.

Formatting conventions shall be applied consistently throughout a document set by the use of style sheets.

If the application uses icons or graphical symbols, the electronic information for users shall explain what they represent. The information for users should use the same icons to represent the same objects. The information for users shall not use those icons to represent other objects.

Where controls are needed within the electronic information for users, for example for navigation, they should obey the conventions agreed to for the information for users.

Formatting conventions for highlighting information of special importance, such as dangers, warnings, cautions, and notes, shall be applied consistently throughout the document set.

The information for users may use special formatting to identify new or changed content. Similar material, such as sets of instructions, shall be presented in a consistent manner.

If information for users is adapted for use in another operating environment, language, or culture, a common glossary and style guides for text and illustrations should be used to assist information developers and translators in maintaining consistency. Consideration should be given to selecting graphics and colours that are not culture-specific, to enable information for users to be more easily adapted, localized, or translated while preserving the intended meaning of the original.

Onscreen information for users should be consistent with the “look-and-feel” format of one of the following:

a) the software being described;

b) other software products in the same suite, or the parent software product of the suite;

c) the operating system;

d) other onscreen information for users with which the user is familiar.

9.3 Selection of appropriate media and format

9.3.1 Comparison of formats and media

A range of factors determine which formats and media are most effective for presenting information. These factors include:

- audience characteristics and information needs;
- nature of tasks that are the subject of the information for users;
- context of use;
- volume of information;
- cost of development and maintenance;
- usability;
- marketability.

The information developer should work within the appropriate constraints to give the user the most convenient information for users possible.

The context of use should influence the information format and media, for example:

- Is the information needed only temporarily, or should it remain visible (persistent)?
- Is the information dynamic or static?
- Does the information have to be available at the same time as the user is working with the application?
- How quickly does the user need an answer to be provided?
- Does the information need to be available when the application is not available, such as information about finding help when the software product does not run?

The information developer should record the source data and design decisions for each audience/task combination in an information profile (Table 5).

Table 5 — Sample information profile for one task and one audience
Task: Get a quotation for a journey Audience: Ticket agent

Information needed	Volume	Same time as the application	Persistent or temporary	Urgency	Medium
Task instructions for getting a quotation including finding and entering departure and destination locations; entering dates, party sizes, and ticket types; checking availability; and finding a price	Up to 10 task steps	Yes	Persistent	Quick	Onscreen list of topics, or menu
Reference list of location codes	Hundreds	Yes	Temporary	Quick	Context-sensitive help
Reference list of ticket types	Up to 50	Yes	Temporary	Quick	Context-sensitive help
Reference list of passenger categories	Maximum of 10	Yes	Temporary	Quick	Context-sensitive help
Field use information for date formats	One line	Yes	Temporary	Quick	Embedded help

Often the solution involves the use of multiple formats and media. The information designer should consider formats and media that are available or can be invented to suit the particular project.

9.3.2 Use of printed or electronic information for users

If a system becomes unusable, printed information for users is the only help available to the user. Whether or not electronic information for users is provided, the following are examples for providing information for users in printed form:

- instructions for starting the software;
- information for performing recovery actions available to users.

The supplier should provide this information in printable format on a website, so that the user can print this information before downloading or installing the software. For software packaged and delivered in a box, this information shall be printed on the box or printed and included in the box. When electronic information for users is provided, it shall be available for display whenever user input to the software is possible. The user should be able to perform a function and get the relevant function-specific electronic information for users simultaneously.

9.3.3 Relationship of information displays to the application's displays

The presentation of all controls in the onscreen information for users should be consistent with the controls for the application or the system. However, onscreen information for users should be presented so that users may distinguish when they are viewing the application and when they are viewing the information for users.

NOTE The ISO 9241 series defines the ergonomics of human-system interaction. ISO 9241-112:2019 gives detailed guidance on presenting information on a screen.

Information that is needed at the same time as the application interface should be displayed in at least one of the following ways:

a) provide the information as embedded information for users;

b) display the information and the application so both are visible at the same time;

c) provide a simple method of switching between the two;

If onscreen reference information for users is provided, it shall be accessible from the software it describes, and shall provide a clear means of exiting the information for users and returning to the software. Software may be linked to online help, tutorials, or reference information for users in various ways, such as the following:

- through a help menu with an entry to the help system;
- through help buttons providing context-sensitive information for a particular user interface or situation;
- through tool tips.

If information is needed all the time, it should not be closed when the user interacts with the user interface.

If information is needed only temporarily, the display window should close when the user takes the next action with the application, so that the user does not have to take special steps to remove it.

9.4 Context-sensitive information

Information for users may provide context-sensitive information, for example:

- the current user interface control;
- the current task;
- the current application function (such as a dialog box, a transaction, or a command);
- the current message.

If it is not possible to determine the context for the help the user needs, one topic should be displayed, and facilities should be provided for the user to choose a different topic.

If the information is context-sensitive:

a) display an item specific to the context;

b) display the relevant information at the top of the information area so that it is the first information users see, if the required information is a piece of some longer topic.

Table 6 provides some examples of access methods suitable for different types of context-sensitive information.

Table 6 — Examples of access methods

Types of information	Access method
Task description for the current task. Function description for the current function. Explanation of the current message.	Press a special key. Click on a help button or icon. Ask for help via voice input.
Names of icons. Uses of icons. Uses of fields. Definitions of terms.	Position the pointer over an object on the screen. Select an object on the screen, for example highlight, click on, or touch the object, using a different technique from that for activating the object.

Table 6 *(continued)*

Types of information	Access method
Process descriptions.	Choose from a menu,
Concepts.	Find from a contents list (Table of Contents)
Exploitation information.	Find via search.
Frequently asked questions.	Find from FAQ list,
Overview of the application.	Ask for help via voice input.
Overview of the information for users.	

9.5 Accessible information for users

9.5.1 Understandable information for users

The term 'onscreen' should only be used when the information is displayed on a screen. It should not be used as a synonym for digital or electronic information, as information can also be input and output via speech.

Product information for users, whether print or electronic, should use a clear and simple language using the vocabulary of the task.

The use of technical terms is acceptable where they are required to clearly explain the functionality or product.

EXAMPLE The information for users of a CAD (computer-aided design) system can use terminology from the field of technical drawing.

9.5.2 Information for users in accessible electronic form

If a person can use the product, the information shall be provided in a form accessible to them.

9.5.3 Text alternatives for non-text information

Information presented electronically in pictures, graphics, video, audio shall also be provided as descriptive text suitable for screen reading, printing, or Braille conversion so that it can be read by an alternative method.

9.5.4 Unnecessary device references

Instructions for software, whether print or electronic, should be written so that they refer to the users' actions and resulting output without reference to a specific device.

For contexts where operation of a specific device such as a mouse is required, it is possible that a generic description cannot be written. However, such specific descriptions need only occur in help about using that device, not in all contexts.

EXAMPLE 1 The task description in help does not require a user to recognize the colour of a user interface control to use it, so the text does not state "click on the green icon". Instead, the name is reported.

EXAMPLE 2 An application provides a description of how to perform tasks using as many different input/output modalities as are available (e.g. mouse, keyboard, voice).

9.5.5 Information on accessibility features

Information for users shall provide general information on the availability of accessibility features and information about the purpose of and the method of using each feature.

NOTE It is important for users to be able to easily discover the accessibility features of the software.

EXAMPLE 1 Online help provides a section describing features of interest for people who have disabilities.

EXAMPLE 2 Online help explains keyboard-only use of the software.

EXAMPLE 3 Online help describes how to adjust the font size.

EXAMPLE 4 A product has multiple colour schemes, and information for users describes which colour schemes are available for people who have difficulties with colour vision.

9.6 Layout of screens and pages

9.6.1 Display area

The information developer shall prepare a consistent layout for formatting similar information.

The information developer should confirm that the information for users is scalable and legible on the devices likely to be used. The information developer should prepare a display area for each type of display to show how the elements required should be positioned. For example, for online help on a desktop screen, the display area should include:

a) the title of the topic;

b) window controls (such as close window);

c) navigational controls (such as scroll bars);

d) information area;

e) non-scrolling and scrolling areas;

f) icons or graphical symbols indicating where the current information is within the topic;

g) placement of annotations, if available;

h) positioning of illustrations.

The design should use consistently named and positioned navigational controls on displays of the same type. The information developer should plan what happens to the information area, the size of text and illustrations, and non-scrolling areas when the window is resized.

9.6.2 Non-scrolling areas

The design should continue to display the following reference information in a non-scrolling area:

a) topic title (in a web browser window this may be displayed in the title bar);

b) links for returning to the main topic or home page;

c) navigational controls for moving to related topics.

NOTE ISO/IEC/IEEE 23026 provides detailed guidance on presentation of technical information through web pages.

9.7 Legibility

9.7.1 General

Printed and onscreen information for users shall be legible to the user, taking into consideration the distance between the user and the information for users in the expected work environment.

Information for users shall use a font style and colour that is legible against the expected background (paper colour or screen background colour).

In languages that use an alphabet with upper- and lower-case letters, upper-case (all capital) letters should not be used for continuous text of more than one sentence, except where legally required. Text in all upper case is difficult to read, and consequently the user often ignores it.

Printed text, including text in illustrations, shall be no smaller than 2,75 mm for capital letters measured at the page. The designer should not rely on internal settings of point sizes as the measure of text displayed on the screen but should observe the sizes of displayed text on all types of screens that can be used.

NOTE IEC/IEEE 82079-1:2019, Table 4 contains examples of minimum point size for various situations.

Onscreen information for users should be legible in the users' expected work environment, which includes the anticipated combination of computer monitor or display and software graphics drivers. Legibility can be affected by output devices (screen displays and printers) that are monochrome, have limited resolution, render colours differently, or support a limited range of colours. Some output devices apply substitute fonts or special characters if the specified font or character set is not available.

Onscreen information on mobile devices like tablets and phones should adopt the fonts and colours from the standard settings of the mobile operating system, so that the legibility is consistent throughout the application.

Distinctions that depend on more than two gradations of colour or shades of grey can be undetectable. Because some users cannot distinguish between colours, information for users should provide text cues rather than using colours such as red or green as the only way to convey meaning. Iconic cues may also be used to support coloured text.

9.7.2 Typefaces and text size

The designer should select typefaces that are readily available for the planned presentation technology or reproduction method. When selecting typefaces and sizes to use for onscreen information for users, the designer should take into account:

a) the range of different display screens that can be used and the users' ability to resize the text;

b) the range of default typefaces and sizes of users' systems;

NOTE 1 If the information for users uses fonts that are not installed on a user's computer, the computer can replace those fonts with default fonts.

c) the different physical environments in which the information for users is used;

d) translation of the information for users: the required character sets for the required languages need to be available;

e) availability of special characters (such as the copyright symbol) that are needed for the information for users;

f) the character sizes in the selected font.

NOTE 2 The same nominal font size can appear quite different in different typefaces.

Check the legibility of planned type fonts on different display types. The design should not use typefaces in which the lower-case letter l, the capital (upper case) letter I, and the number 1 (one) are indistinguishable, or those in which the number 0 and the capital (upper case) letter O are indistinguishable.

9.7.3 Highlighting text

Italic text can be difficult to read, particularly on a screen, for more than a few words. Italic text may be used for the following:

— command variables;

- introduction of new terms;
- references to article and book titles.

Underscored text should be used only to represent active hyperlinks. In onscreen information for users, the link should also change colour when selected.

The following methods of highlighting should be avoided as visually distracting for the user:

a) underlining that crosses the descenders of characters;

b) blinking and animation (words should not be highlighted by blinking the text itself; if necessary, a marker, which may blink, should be displayed next to the text);

c) complete words or phrases in upper-case (all capital) letters.

9.8 Formats for representing user interface elements

9.8.1 General

Graphical user interface (GUI) elements of the software or the application, such as buttons, icons, cursors and pointers; special uses of keyboard keys or combinations of keys; uses of mouse clicks; uses of touches on touch screens; and system responses shall be represented in information for users by consistent graphic or typographical formats so that the various elements are each distinguished from the text. The information for users should include a representation of the element, its purpose, and an explanation of its action (functional consequence), with examples of actual operational instances. Onscreen information for users may include pop-up text labels for GUI elements.

Information for users incorporating text from software dialogs should copy the text exactly, character by character.

9.8.2 Representing control and command input

Formats for user-entered commands or codes shall clearly distinguish between literals (to be input exactly as shown) and variables (to be provided by the user).

EXAMPLE The variable "filename" can be used as a general term for which a real filename has to be substituted.

Information for users should represent variables in a distinctive type (e.g. italics).

Formal notation, such as the use of brackets, braces, greater than (>) and less than (<) characters, and other marks, shall be defined in every piece of information for users that uses it, with the exception of [...] for references in a bibliography. Quotation marks shall not be used in command representations unless the user should input them literally.

9.8.3 Representing special keyboard keys or mouse clicks

The information for users shall establish consistent conventions for representing special keyboard keys and explain them. In text, the information for users should distinguish terms or symbols referring to individual keyboard keys from the same characters used in other contexts with a convention, such as one of the following:

- a special type style or font; for example, use of bold characters;
- pictures of key tops; if the pictures are larger than the normal text of the information for users, the pictures may be placed in the margin, in a list, or in a table;
- special characters such as < and > to enclose the term or symbol. Check carefully that those characters themselves do not need to be represented in the same way.

Information for users should address variations in keyboards or data entry devices in the expected users' work environment. Information for users should indicate whether different keyboards, with different characters on the key tops, may be used. The information for users should set consistent conventions for the names of keys such as Enter, Shift, and Tab. Instructions and help for software should be written so that they refer to the users' actions and resulting output without reference to a specific device. References to devices, such as the mouse or the keyboard, should only be made when they are integral to and necessary for understanding of the advice being given.

A convention is frequently needed for representing a space. The information for users should use a character not used for other purposes in either the information for users or the software.

9.8.4 Representing interactions on touch screens

For touch screens the interactions such as taps, double taps, touches and gestures should be represented in an appropriate manner. The terms used for touches should be suitable for that kind of interaction (e.g. “touch the object” or special terms like “swipe” or “zoom with two fingers”).

Alternate techniques to interaction with fingers, such as a pencil, shall be taken into consideration. Illustrations of touches should use realistic pictures of the movement of fingers on the touch screen. Animations of touch are recommended for explaining more complicated or special gestures.

9.9 Use of colour

Although colour is a valuable method of conveying meaning, the designer should consider whether colour is essential to the information for users. In some information for users, such as graphics or visualization software, exact representations of colour are indispensable. In most other information for users, colour is used to emphasize keywords or safety-related information. Because software may also be used by people who have difficulties with colour vision, the design should not use colour alone to convey meaning; but should employ another form of highlighting as well, such as different types of hatching or shading patterns in illustrations.

When using colour, the designer should:

a) determine the colour palettes available on the users' display devices. Use only colours from the standard palette, to display colours correctly on users' screens;

b) select a neutral background colour because the perceptibility of objects depends on background colour. The designer should take care when placing coloured objects on a non-white background. Use solid colour (such as white) for background or to fill areas of the display.

c) determine whether the users are likely to print the information for users on monochrome printers. In particular note that the colours selected using the RGB gamut available on a computer monitor or an inkjet printer can look different when printed on a device that uses the CMYK gamut (offset or colour laser technologies);

d) where consistency of colour is important between the embedded and printed information for users, check the proposed colours thoroughly;

The standardized colours specified in ISO 3864-1 for use in safety signs should not be used for any other purpose in information for users related to software.

9.10 Navigational features

9.10.1 General

Information for users shall contain features to provide access to information, such as a table of contents, an index, and search capabilities.

Navigational features should indicate the locations to which users can move from their current location. If facilities are provided to link to topics in different parts of the hierarchy, the information for users should display where the topic is in the hierarchy. Where users move between topics in electronic information for users, the information for users should indicate where the current topic fits into the total structure or may contain a diagram explaining where the current topic belongs in the structure, or highlight the new position in a contents list.

Information for users should provide a simple method of exiting from the electronic information. The exit method should be the same throughout the electronic information.

9.10.2 Finding the same information again

Features for navigation shall be provided such that users can determine their location within the printed or electronic information for users.

Printed or printable PDF information for users shall include consecutive page numbers. Page numbering may be in one of the following formats:

a) consecutive throughout with Arabic numbers; optionally, small Roman numerals may be used for the front matter;

b) consecutively numbered within each section;

c) consecutively numbered in the main text with separate numbering for each annex.

Figures and tables should each be labelled consecutively throughout the document or section.

Other navigational aids should be provided for ease of reference, such as distinctive edge tabs or page headers identifying each section.

For electronic information for users, the navigational features should remain accessible, e.g. if onscreen information for users allows scrolling.

Information for users should enable the user to return to information later in the present session or in another session. This return may be accomplished by these techniques:

— a cumulative list of the topics the user has looked at either in this session or in this and previous sessions, from which they may select topics;

— bookmarks that enable users to mark places to which they want to return;

— a clear structure that allows users to follow the same paths;

— annotated or highlighted topics that show which topics the user has selected.

Techniques used should be consistent and simple to use.

9.10.3 Formats for active areas

Interactive text of different types should be clearly distinguished, so that users understand what happens if they select the text. The following should be distinguished:

a) active areas of text causing a link to another topic, replacing the current topic;

b) active areas of text causing additional information to be displayed as well as the current topic (usually in a separate or pop-up window or as expanded inline or drop-down text).

NOTE Information displayed in a pop-up is not usually printable for future reference.

Interactive illustrations can be used for image mapping. Different regions of the illustration are sensitive hot-spots containing onscreen information for users or link to a target.

EXAMPLE 1 A flowchart has several blocks. Clicking on a block links to a relevant concept topic.

EXAMPLE 2 A screen display of the GUI is mapped in regions such as navigation bar, menu line, working area. Clicking on a region opens a popup with a short description.

As the function of an active area on an illustration is not obvious for first sight, it is necessary to give a hint to the interactivity by a caption like "Areas in the graphic are clickable".

The designer should select a distinct method for representing each type of active area of text, such as one of the following.

- The shape of the pointer may change as it moves over the active area.
- The active areas may be indicated by using different methods of highlighting or different methods of displaying the area.

Text marked with a solid underscore should be used only to represent active hyperlinks. The hyperlink should also change colour when selected.

9.10.4 Linking information

Links should provide information that the user expects in one jump, rather than requiring that the user follows one or more additional links to reach the required information.

If the destination is outside the particular suite of information for users, the users should be notified that they are leaving it. The information for users should provide users with an alternate way of locating the information, in case the external link has been broken or the destination removed.

Links between related topics shall be bidirectional, so that whichever topic the users access first, they may jump to the related information on the other topic.

Links shall provide a clear indication of the destination of the link. For example, use
"More troubleshooting tips" rather than "Click here."

If facilities are provided for the user to link by selecting active areas of text, the information for users should distinguish this text from the surrounding text and active areas of text used to obtain clarification. Clarification elements should be collected at the start or end of topics under headings such as "Related topics" or "See also."

9.10.5 Table of contents

Headings and page numbers appearing in the table of contents shall be the same as those used in the text. The table of contents should have enough levels to help the target audience to find information efficiently, but not exceed four levels of headings. Lengthy documents may have a summary table of contents at the beginning and a more detailed table of contents for each section. Lists of Figures and Lists of Tables should also be included if there are more than a few illustrations.

Onscreen information for users may display tables of contents in expandable and collapsible formats (tree structure) to provide top-level and detailed access to headings without excessive scrolling.

Secondary, comprehensive tables of contents may be accessible through menus, expandable lists, or secondary windows. To expedite navigation to the desired topic in a hierarchy of menus, the designer should carefully allocate the number of entries to show at each menu level and the number of levels of menu. To reduce the number of levels, the design should offer more choices at each level or display several levels at once on each menu.

In onscreen information for users, the table of contents may be displayed in its own navigation window, or as a separate frame in the information window. The table of contents should remain accessible while the user is selecting and reading topics.

9.10.6 Index

An index is necessary in printed information for users if it is lengthy and complex. Onscreen documents or help systems shall include a search tool (9.10.7) and, optionally, an index whose access points are electronic links.

The following is guidance on good practices for indexing information for users.

a) Indexes should include alternative terminology from that used in the information for users. To determine the different terms users may choose, consider why they want to look up the topics. Use synonyms to provide alternate terms for access.

b) The preface, the appendixes, dangers, warnings, cautions, and notes should be included. Non-textual items such as graphics (and multimedia in onscreen information for users) should be indexed.

c) Overly general keywords and user-descriptive terms for entries should be avoided, and the main word should be placed first. For example, for the text heading, “using file manager,” use the index heading, “file manager, using.”

d) Entries should be double-posted. For example, a topic that is indexed as both “site licenses” and “licenses, site” is double-posted.

NOTE More information about indexing can be found in ISO 5963.

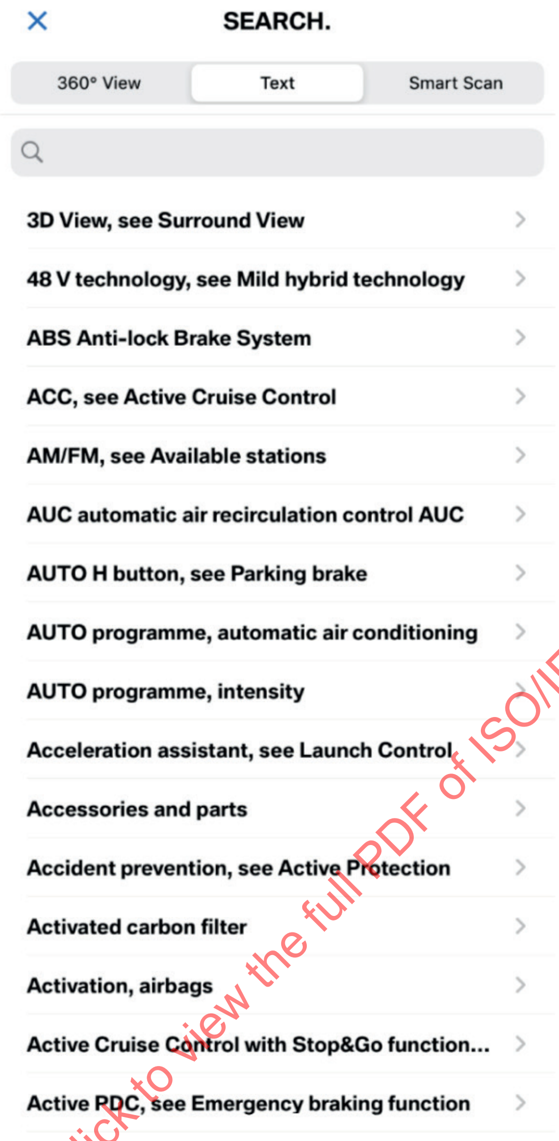
9.10.7 Search capability

Onscreen information for users shall provide a method of locating words in the entire text. Electronic search capabilities should be configured with a maximum of functions and options, such as:

- full text search of the information for users;
- search for topic titles only;
- ranked returns from the search indicating their closeness to the requested topic;
- search for words in illustrations;
- search with predefined filters, categories, or software objects such as menu objects or a parameter;
- text string in the current topic;
- Boolean search (a search in which search words are logically combined by the use of the logical operators OR, AND, NOT);
- type-sensitive search so that the results are updated type by type;
- text string search using a regular expression.

Search for common, non-meaningful words (such as "the" in English) should be turned off.

Search functions can be applied to terms that would ordinarily be in an index. In this situation, the alphabetic index keywords are presented in the search results even if no search item is entered (Figure 5).



3D View, see Surround View
48 V technology, see Mild hybrid technology
ABS Anti-lock Brake System
ACC, see Active Cruise Control
AM/FM, see Available stations
AUC automatic air recirculation control AUC
AUTO H button, see Parking brake
AUTO programme, automatic air conditioning
AUTO programme, intensity
Acceleration assistant, see Launch Control
Accessories and parts
Accident prevention, see Active Protection
Activated carbon filter
Activation, airbags
Active Cruise Control with Stop&Go function...
Active PDC, see Emergency braking function

Figure 5 — Example of an index search

A quick access to the search can be provided by a search field for the information for users that is located in the software menu bar itself. This search field should be placed next to the help menu in the menu bar or as a menu item in the help menu (Figure 6). Starting the search function opens the help system and shows the results. To realize this Search feature, the information developer and the software developer should work together.

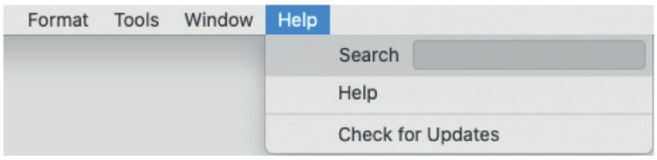


Figure 6 — Example of a search field in the help menu

9.11 Format of danger, warning, and caution indications

Requirements for danger, warning, and cautions indications related to software are stated in IEC/IEEE 82079-1:2019, 9.15.1: