INTERNATIONAL STANDARD

ISO/IEC 25010

Second edition
2023-11

Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model

Ingénierie des systèmes et du logiciel — Exigences de qualité et évaluation des systèmes et du logiciel (SQuaRE) — Modèles de qualité du produit

Reference number
ISO/IEC 25010:2023(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.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 document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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.

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

This second edition of ISO/IEC 25010, together with the first edition of ISO/IEC 25002 and the first edition of ISO/IEC 25019, cancels and replaces ISO/IEC 25010:2011, which has been technically revised.

The main changes are as follows:

— This document revises the product quality model part of ISO/IEC 25010:2011. The other parts are moved to ISO/IEC 25002 on quality models overview and usage and ISO/IEC 25019 on quality-in-use model. The quality characteristics and subcharacteristics of the product quality model are revised for the purpose of better understanding and fitting the state of the art of ICT (information and communication technology).

— The target of the product quality model has been extended to include various types of ICT product and information system.

— Safety has been added as a quality characteristic with subcharacteristics, i.e. operational constraint, risk identification, fail safe, hazard warning and safe integration.

— Usability and portability have been replaced with interaction capability and flexibility respectively.

— Inclusivity and self-descriptiveness, resistance, and scalability have been added as subcharacteristics of interaction capability, security, and flexibility respectively.

— User interface aesthetics and maturity have been replaced with user engagement and faultlessness respectively.

— Accessibility has been split into inclusivity and user assistance.

— Several characteristics and subcharacteristics have been given more accurate names and definitions.

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

ICT (information and communication technology) products, including software products, are increasingly used to perform a wide variety of organizational and personal activities. Realization of goals and objectives for personal satisfaction, organizational success and/or human safety relies on high-quality ICT products. High-quality ICT products are essential to providing value and avoiding potential negative consequences for the stakeholders. The term "product" is used for ICT products which can include software, data, hardware and communication facilities, and other ICT products throughout this document. A product has a variety of influences on many classes of stakeholders including those who develop, acquire, and use the product. Stakeholders also include customers of businesses using the product, as well as the public under the influence of information systems using the product under real operation.

A comprehensive specification and evaluation of the target product is a key factor in ensuring value to stakeholders. This can be achieved by defining the necessary and desired quality characteristics associated with the stakeholders' goals and objectives for the system. This includes quality characteristics related to the product and data as well as the impact the system has on its stakeholders. It is important that the quality characteristics be specified, measured, and evaluated whenever possible using validated or widely accepted measures and measurement methods. The quality model in this document can be used to establish requirements, their criteria for satisfaction and the corresponding measures. A comparison with the product quality model in ISO/IEC 25010:2011 is given in Annex A.

This document is intended to be used in conjunction with the other documents in the SQuaRE family of International Standards (ISO/IEC 25000 to ISO/IEC 25099).

This document is a part of the SQuaRE family of International Standards. Figure 1 illustrates the organization of the SQuaRE family of International Standards. Similar standards are grouped into divisions. Each division provides guidance and resources for performing a different function in ensuring system and software product quality. This document belongs to the quality model division and is aligned with ISO/IEC 25002 belonging to the quality management division.

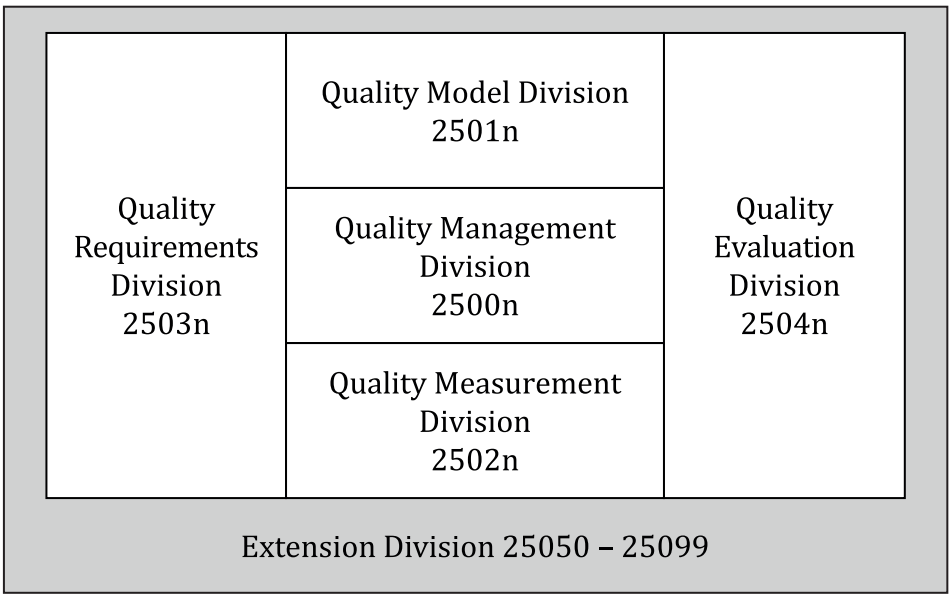


Figure 1 — Organization of SQuaRE family of International Standards

The divisions within the SQuaRE family are;

— ISO/IEC 2500n - quality management division. The International Standards that form this division define all common models, terms, and definitions referred to by all other International Standards from the SQuaRE family. This division also provides requirements and guidance for a supporting

function that is responsible for the management of the requirements, specification, and evaluation of software product quality. Practical guidance on the use of the quality models is also provided.

  - ISO/IEC 25000: Guide to SQuaRE
  - ISO/IEC 25001: Planning and management
  - ISO/IEC 25002: Quality models overview and usage

- ISO/IEC 2501n - quality model division. The International Standards that form this division present detailed quality models for computer systems and software products, data, IT services and quality-in-use.
  - ISO/IEC 25010: Product quality model
  - ISO/IEC TS 25011: Service quality models
  - ISO/IEC 25012: Data quality model
  - ISO/IEC 25019: Quality-in-use model
- ISO/IEC 2502n - quality measurement division. The International Standards that form this division include a quality measurement framework, mathematical definitions of quality measures, and practical guidance for their application. Examples are given of quality measures for internal and external property of product, data, IT services and quality-in-use. Quality measure elements (QME) forming foundations for quality measures for internal and external property of product are defined and presented.
- ISO/IEC 2503n - quality requirements division. The International Standards that form this division help specify quality requirements based on quality models and quality measures. These quality requirements can be used in the process of eliciting quality requirements for information systems and IT services to be developed or as input for an evaluation process.
- ISO/IEC 2504n - quality evaluation division. The International Standards that form this division provide requirements, recommendations and guidelines for software product evaluation, whether performed by evaluators, acquirers or developers. The guideline for documenting a measure as an evaluation module is also provided.
- ISO/IEC 25050 to ISO/IEC 25099 - SQuaRE extension division. These International Standards currently include requirements for quality of ready-to-use software product (RUSP) and instructions for testing, Common Industry Format (CIF) for usability reports, and quality models and measures for new technologies such as cloud services and artificial intelligence.

The SQuaRE standards can be used in conjunction with ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288, particularly the processes for the specification and evaluation of quality requirements. ISO/IEC 25030 describes how quality models can be used for systems and software quality requirements; and ISO/IEC 25040 describes how the quality models can be used for systems and software quality evaluation.

The SQuaRE standards can also be used in conjunction with ISO/IEC 33000 family of International Standards which are concerned with software process assessment to provide:

- a framework for software product quality definition in the customer-supplier process;
- support for quality review, verification, and validation, as well as a framework for establishing quantitative quality characteristics;
- support for setting organizational quality goals in the management process.

The SQuaRE standards can be used in conjunction with ISO 9001 (which is concerned with quality assurance processes) to provide:

- support for setting quality goals;

— support for design review, verification, and validation.

Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model

1 Scope

This document defines a product quality model, which is applicable to ICT (information and communication technology) products and software products. The product quality model is composed of nine characteristics (which are further subdivided into subcharacteristics) that relate to quality properties of the products. The characteristics and subcharacteristics provide a reference model for the quality of the products to be specified, measured and evaluated.

NOTE 1 In this document, a product refers to an ICT product that is part of an information system. ICT product components include subsystems, software, firmware, hardware, data, communication infrastructure, and other elements that are part of the ICT product.

This model can be used for requirements specification and evaluation of the target products' quality throughout their lifecycle by several stakeholders, including developers, acquirers, quality assurance and control staff and independent evaluators. Activities in the product lifecycle that can benefit from the use of this model include:

— eliciting and defining product and information system requirements;

— validating the comprehensiveness of requirements definition;

— identifying product and information system design objectives, and design necessary process for achieving quality;

— identifying product and information system testing objectives;

— identifying quality control criteria as the part of quality assurance;

— identifying acceptance criteria for a product and/or an information system;

— establishing measures of product quality characteristics in support of these activities.

NOTE 2 Usage of the quality model for measurement is explained in Annex C.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

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

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

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

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

3.1
functional suitability
capability of a product to provide functions that meet stated and implied needs of intended users when it is used under specified conditions

Note 1 to entry: Functional suitability is concerned with whether the functions meet not only stated and implied needs, but also the functional specification (see C.1).

3.1.1
functional completeness
capability of a product to provide a set of functions that covers all the specified tasks and intended users' objectives

3.1.2
functional correctness
capability of a product to provide accurate results when used by intended users

Note 1 to entry: Precision is one of the attributes of correctness.

EXAMPLE In case of the products requiring high precision such as scientific software, the product can provide precise results with the needed degree as well as accurate results.

3.1.3
functional appropriateness
capability of a product to provide functions that facilitate the accomplishment of specified tasks and objectives

EXAMPLE A product provides the necessary and sufficient steps to complete a task, excluding any unnecessary steps.

Note 1 to entry: Functional appropriateness corresponds to suitability for the task in ISO 9241-110.

3.2
performance efficiency
capability of a product to perform its functions within specified time and throughput parameters and be efficient in the use of resources under specified conditions

Note 1 to entry: Resources can be CPU, memory, storage, and network devices.

Note 2 to entry: Resources can include other software products, the software and hardware configuration of the system, energy, and materials (e.g. print paper, storage media).

3.2.1
time behaviour
capability of a product to perform its specified function under specified conditions so that the response time and throughput rates meet the requirements

3.2.2
resource utilization
capability of a product to use no more than the specified amount of resources to perform its function under specified conditions

3.2.3
capacity
capability of a product to meet requirements for the maximum limits of a product parameter

Note 1 to entry: Parameters can include the number of items that can be stored, the number of concurrent users, the communication bandwidth, the throughput of transactions, and the size of a database.

3.3
compatibility
capability of a product to exchange information with other products, and/or to perform its required functions while sharing the same common environment and resources

3.3.1
co-existence
capability of a product to perform its required functions efficiently while sharing a common environment and resources with other products, without detrimental impact on any other product

3.3.2
interoperability
capability of a product to exchange information with other products and mutually use the information that has been exchanged

Note 1 to entry: Information is meaningful data; and information exchange includes transformation of data for exchange.

3.4
interaction capability
capability of a product to be interacted with by specified users to exchange information between a user and a system via the user interface to complete the intended task

Note 1 to entry: Interaction capability in the product quality model and its subcharacteristics focus on a set of attributes that enable interaction by users (or operators) to complete specific tasks in a variety of contexts of use. On the other hand, usability as defined in the quality-in-use model (ISO/IEC 25019) comprehensively focuses on outcomes of use to determine whether tasks are achieved by users with effectiveness, efficiency and satisfaction in a specific context of use.

Note 2 to entry: Interaction capability is a prerequisite for usability.

Note 3 to entry: Interaction itself is defined in ISO TR 25060 as “exchange of information between a user and an interactive system via the user interface”.

3.4.1
appropriateness recognizability
capability of a product to be recognized by users as appropriate for their needs

Note 1 to entry: Appropriateness recognizability depends on the ability to recognize the appropriateness of the product functions from initial impressions of the product or system and/or any associated documentation.

Note 2 to entry: The information can be provided by the product to assist users in making decisions about the adoption, acquisition, or use of products prior to the start of full-scale use, through demonstrations, tutorials, documentation or, for a website, the information on the home page.

3.4.2
learnability
capability of a product to have specified users learn to use specified product functions within a specified amount of time

3.4.3
operability
capability of a product to have functions and attributes that make it easy to operate and control

Note 1 to entry: Operability is related to controllability, user error robustness and conformity with user expectations as defined in ISO 9241-110. It is also related to the effectiveness and efficiency of physical interface devices (e.g. mouse, touch pen).

3.4.4
user error protection
capability of a product to prevent operation errors

3.4.5
user engagement
capability of a product to present functions and information in an inviting and motivating manner encouraging continued interaction

Note 1 to entry: This refers to properties of the product that increase the pleasure and satisfaction of the user, such as harmonious colour, intuitive user interface and friendly voice guidance.

3.4.6
inclusivity
capability of a product to be utilised by people of various backgrounds

Note 1 to entry: Backgrounds include (and are not limited to) people of various ages, abilities, cultures, ethnicities, languages, genders, economic situations, education, geographical locations and life situations.

3.4.7
user assistance
capability of a product to be used by people with the widest range of characteristics and capabilities to achieve specified goals in a specified context of use

Note 1 to entry: The range of capabilities includes language differences and disabilities associated with age, sight, hearing, use of hands, arms and legs, etc.

Note 2 to entry: A set of specific rules and methods for software accessibility can be applied to ensure "user assistance" in this document and in ISO/IEC 25019.

EXAMPLE A system that can interact with users using multiple input/output methods, such as voice, gaze, and touch, in addition to visual display, in order to accommodate differences in vision, hearing, and body parts that can be moved, or changes in these areas.

Note 3 to entry: Inconveniences or less effectiveness for users caused by differences or changes in their actual contexts of use beyond those initially specified in the requirements can be resolved by repeating the quality improvement cycle to find and resolve problems through iterative evaluation and requirements definition from this quality characteristic perspective.

Such differences or changes in the context of use can include, for example, the following cases:

— when using the system while driving a car or flying in an airplane;

— when using the system interactively for emergency within short-time use and small screen view due to an accident or a disaster;

— when a user is a beginner or is changing own task goal of use, usage, or skill and knowledge;

— when a user having different physical capabilities due to the type of injury or illness, or changes in time due to healing or progression.

Also, other quality (sub)characteristics, typically such as *adaptability* (3.8.1) or *flexibility* (3.8), can be collaboratively applied to improve user assistance and *interaction capability* (3.4).

3.4.8
self-descriptiveness
capability of a product to present appropriate information, where needed by the user, to make its capabilities and use immediately obvious to the user without excessive interactions with a product or other resources

Note 1 to entry: Other resources include user documentation, help desks, other users and other sources of assistance.

EXAMPLE Instructions for user operation are divided and displayed or talked through step-by-step interactively at the helpful timing of operation, in order to help users understand easily what is going on with the system/software and to prevent users from becoming confused by receiving too many instructions at once.

3.5
reliability
capability of a product to perform specified functions under specified conditions for a specified period of time without interruptions and failures

Note 1 to entry: Wear does not occur in software. Limitations in reliability are due to results from faults in requirements, design and implementation, or from contextual changes.

Note 2 to entry: Dependability is often used as a synonym for reliability. However, dependability has a larger scope in that it includes *security* (3.6), *performance efficiency* (3.2), and continuing support and others in addition to the subcharacteristics of reliability as discussed in Annex B.

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.3387, modified — "degree to which a system, product or component performs" has been changed to "capability of a product to perform"; "without interruptions and failures" has been added; the original note 1 to entry has been replaced by two new notes to entry.]

3.5.1
faultlessness
capability of a product to perform specified functions without fault under normal operation

Note 1 to entry: The concept of faultlessness can also be applied to other quality characteristics to indicate the degree to which they meet required needs under normal operation.

3.5.2
availability
capability of a product to be operational and accessible when required for use

Note 1 to entry: Externally, availability can be assessed by the proportion of total time during which the system, product or component is in an up state. Availability is therefore a combination of faultlessness (which governs the frequency of failure), *fault tolerance* (3.5.3) and *recoverability* (3.5.4) (which governs the length of down time following each failure).

Note 2 to entry: Failover or duplication of systems can be applied to support availability.

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.313, modified — "degree to which a system or component is" has been changed to "capability of a product to be"; the original note to entry has been replaced by two new notes to entry.]

3.5.3
fault tolerance
capability of a product to operate as intended despite the presence of hardware or software faults

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.1574, modified — “degree to which a system, product or component operates” has been changed to “capability of a product to operate”.]

3.5.4
recoverability
capability of a product in the event of an interruption or a failure to recover the data directly affected and re-establish the desired state of the system

Note 1 to entry: The length of the unavailable period following a failure, during which a product is not available at the same level of use as before the failure, is determined by its recoverability. However, the recoverability of a product depends on the recoverability of the computer system on which the product operates or a subset of its functions.

3.6
security
capability of a product to protect information and data so that persons or other products have the degree of data access appropriate to their types and levels of authorization, and to defend against attack patterns by malicious actors

Note 1 to entry: As well as data stored in or by a product or system, security also applies to data in transmission.

3.6.1
confidentiality
capability of a product to ensure that data are accessible only to those authorized to have access

3.6.2
integrity
capability of a product to ensure that the state of its system and data are protected from unauthorized modification or deletion either by malicious action or computer error

3.6.3
non-repudiation
capability of a product to prove that actions or events have taken place, so that the events or actions cannot be repudiated later

3.6.4
accountability
capability of a product to enable actions of an entity to be traced uniquely to the entity

[SOURCE: ISO 7498-2:1989, 3.3.3, modified — "The property that ensures that" has been changed to "capability of a product to enable"; "may be" has been changed to "to be".]

3.6.5
authenticity
capability of a product to prove that the identity of a subject or resource is the one claimed

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.302, modified — “degree to which” has been changed to” capability of a product”; the structure of the sentence has been changed accordingly.]

3.6.6
resistance
capability of a product to sustain operations while under attack from a malicious actor

Note 1 to entry: A malicious attack can include a denial of service attack, a ransomware attack, or other malicious actions. Then, the following approaches can be applied to improve the capability of a product for the quality subcharacteristic:

— to continuously protect itself from well-known attacks by removing potential flaws or weaknesses of the product with the use of *security* (3.6) tools such as a security weakness diagnostic tool, vulnerability scanner and static analysis tool;

— to minimize vulnerability of a product with secure software coding and/or by incorporating security enhancement functions or mechanisms;

— to maintain product updates during its life time for security reasons.

3.7
maintainability
capability of a product to be modified by the intended maintainers with effectiveness and efficiency

Note 1 to entry: Modifications can include corrections, improvements or adaptation of the product to changes in environment, and in requirements and functional specifications. Modifications include those carried out by specialized support staff, and those carried out by business or operational staff, or end users.

Note 2 to entry: Maintainability includes installation of updates and upgrades.

Note 3 to entry: Maintainability can be interpreted as either an inherent capability of the product to facilitate maintenance activities, or the quality-in-use experienced by the maintainers for the goal of maintaining the product.

3.7.1
modularity
capability of a product to limit changes to one component from affecting other components

Note 1 to entry: Modularity implies that the product is composed of discrete modules or components with cohesive content and minimal coupling to other modules or components.

3.7.2
reusability
capability of a product to be used as assets in more than one system, or in building other assets

3.7.3
analysability
capability of a product to be effectively and efficiently assessed regarding the impact of an intended change to one or more of its parts, to diagnose it for deficiencies or causes of failures, or to identify parts to be modified

Note 1 to entry: Implementation can include providing mechanisms for the product to analyse its own faults and providing reports prior to a failure or other event.

3.7.4
modifiability
capability of a product to be effectively and efficiently modified without introducing defects or degrading existing product quality

Note 1 to entry: Implementation includes coding, designing, documenting and verifying changes.

Note 2 to entry: *Modularity* (3.7.1) and *analysability* (3.7.3) can influence modifiability.

Note 3 to entry: Modifiability is a combination of changeability and stability.

3.7.5
testability
capability of a product to enable an objective and feasible test to be designed and performed to determine whether a requirement is met

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.4271, modified — "extent to which an objective and feasible test can be" has been changed to "capability of a product to enable an objective and feasible test to be"; "designed" has been changed to "designed and performed".]

3.8
flexibility
capability of a product to be adapted to changes in its requirements, contexts of use, or system environment

Note 1 to entry: Flexibility to context of use should consider two distinguished aspects, i.e. technical and non-technical. The technical aspect is related with the execution environment of products, such as software, hardware and communication facility; and the non-technical aspect is related with the social environment, such as user and task, and the physical environment, such as climate and nature.

3.8.1
adaptability
capability of a product to be effectively and efficiently adapted for or transferred to different hardware, software or other operational or usage environments

Note 1 to entry: Adaptations include those carried out by specialized support staff, and those carried out by business or operational staff, or end users.

Note 2 to entry: If the product is to be adapted by the end user, adaptability corresponds to suitability for individualization as defined in ISO 9241-110.

Note 3 to entry: Ineffectiveness or inefficiencies for users caused by differences or changes in their actual contexts of use beyond those initially specified in the requirements can be resolved by repeating the quality improvement cycle, in which evaluation and requirements definition are iterated from the perspective of this quality characteristic, and problems are discovered and solved.

Also, other quality subcharacteristics, typically such as *inclusivity* (3.4.6) and *user assistance* (3.4.7), can be collaboratively applied to improve adaptability and *flexibility* (3.8).

Such differences or changes in the context of use can include, for example, the following cases:

— when using the system initially unintended circumstances, e.g. underwater or in outer space;

— when using the system in limited operation mode with less energy supplies or no communication network connectivity due to an accident or a disaster;

— when initially unintended types of users are involved or influenced.

3.8.2
scalability
capability of a product to handle growing or shrinking workloads or to adapt its *capacity* (3.2.3) to handle variability

3.8.3
installability
capability of a product to be effectively and efficiently installed successfully and/or uninstalled in a specified environment

Note 1 to entry: If the product is to be installed by an end user, installability can affect the resulting *functional appropriateness* (3.1.3) and *operability* (3.4.3).

3.8.4
replaceability
capability of a product to replace another specified product for the same purpose in the same environment

Note 1 to entry: The replaceability of a new version of a software product is important to the user when upgrading.

Note 2 to entry: Replaceability can include attributes of both *installability* (3.8.3) and *adaptability* (3.8.1). The concept has been introduced as a subcharacteristic of its own because of its importance.

Note 3 to entry: Replaceability reduces lock-in risks, so that other software products can be used in place of the present one, for example by the use of standardized file formats.

3.9
safety
capability of a product under defined conditions to avoid a state in which human life, health, property, or the environment is endangered

Note 1 to entry: In this document, safety is defined to describe capability of product to be able to avoid exposures which is not tolerable. Then, the definition is different from the other standards relating to safety that define safety as freedom from unacceptable risks.

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.48, modified — “expectation that a system” has been changed to “capability of a product”; “does not, under defined conditions, lead to a state” has changed to “under defined conditions to avoid a state”; note to entry has been added.]

3.9.1
operational constraint
capability of a product to constrain its operation to within safe parameters or states when encountering operational hazard

Note 1 to entry: Operational hazard is a hazardous situation, a circumstance in which people, property or the environment are exposed to an unacceptable risk during operation.

EXAMPLE 1 A constraint that prevents an airplane from entering a stall condition caused by environmental conditions or pilot error.

EXAMPLE 2 A constraint that limits the amount of radiation released by a radiology device to a safe threshold regardless of the operator input.

3.9.2
risk identification
capability of a product to identify a course of events or operations that can expose life, property or environment to unacceptable risk

3.9.3
fail safe
capability of a product to automatically place itself in a safe operating mode, or to revert to a safe condition in the event of a failure

Note 1 to entry: When the fail safe quality subcharacteristic is applied to a complex system or software, it is often very difficult to determine which behaviour is safer. In such a case, functional *safety* (3.9) concept can be used and then, hazard analysis and safety risk assessment can be conducted to derive safety goals and safety requirements to be achieved.

Note 2 to entry: The following approaches can be applied to improve the capability of a product for the quality subcharacteristic:

— to operate in a manner that eliminates or minimizes damage caused by a failure;

— to implement a mechanism to transfer control of a sufficient set of operations to continue safe performance when a failure occurs.

EXAMPLE A traffic light that reverts to blinking red in all directions when normal operation fails.

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.1558, modified — "pertaining to" has been changed to "capability of"; "a system or component that" has been changed to "a product to"; "or to revert to a safe condition" has been added; the notes to entry and EXAMPLE have been added.]

3.9.4
hazard warning
capability of a product to provide warnings of unacceptable risks to operations or internal controls so that they can react in sufficient time to sustain safe operations

EXAMPLE A pedestrian traffic light gives a warning sign, such as showing the remaining seconds, before reverting from green to yellow or red.

3.9.5
safe integration
capability of a product to maintain *safety* (3.9) during and after integration with one or more components

4 Product quality model

4.1 Product quality model structure

The product quality model categorizes product quality properties into nine characteristics: functional suitability, performance efficiency, compatibility, interaction capability, reliability, security, maintainability, flexibility and safety. Each characteristic is composed of a set of related subcharacteristics (Figure 2).

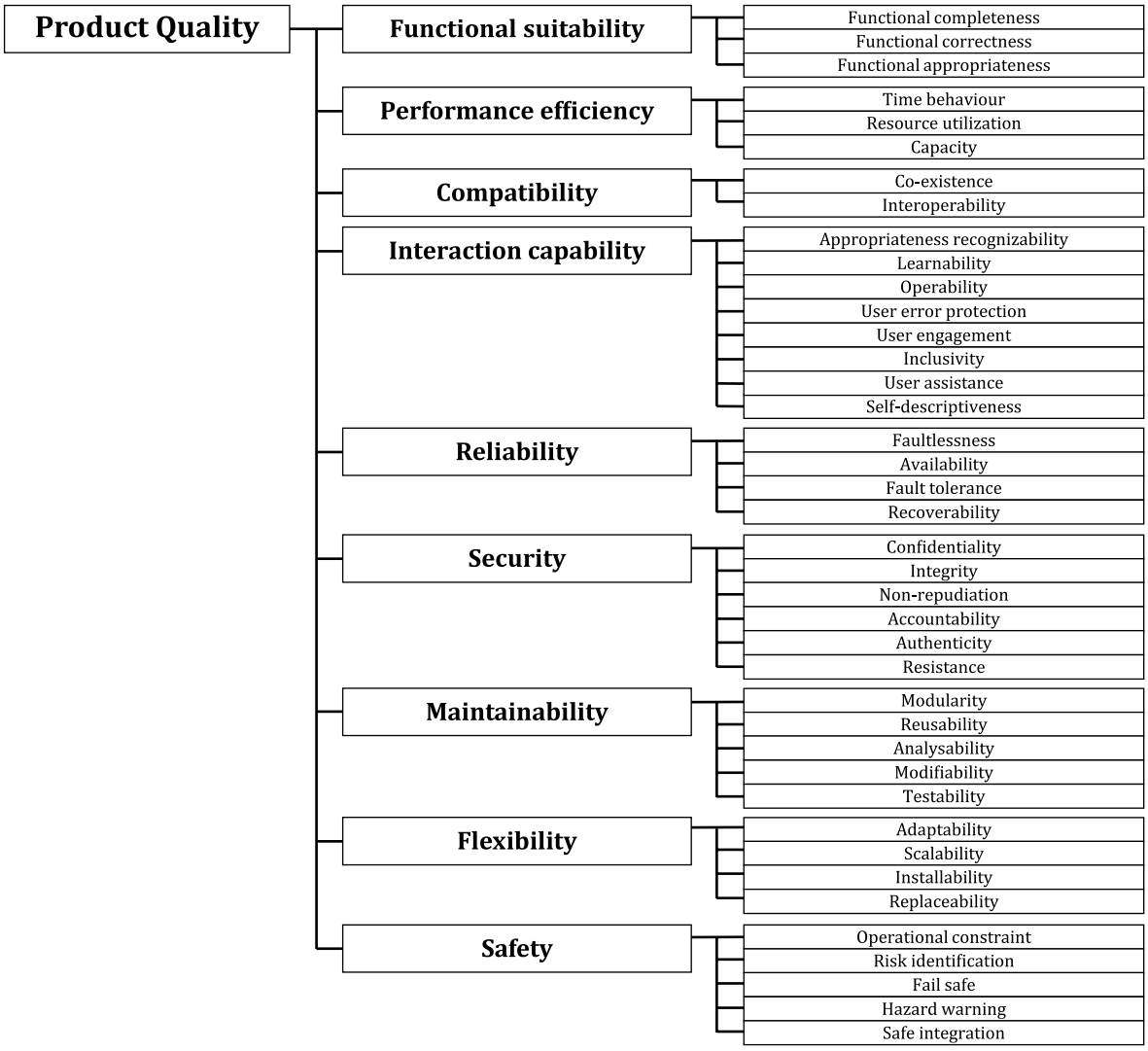


Figure 2 — Product quality model

4.2 Targets of the product quality model

Figure 3 depicts how different quality models apply to different components of an information system or IT service system. The quality models together serve as a framework to ensure that all relevant aspects of quality are considered in specifying requirements, developing, or evaluating information systems, ICT products, data, or IT services. However, the priorities assigned to the quality characteristics in each model depend on the context in which the information system is used. The context of use represents a collection of interacting factors including:

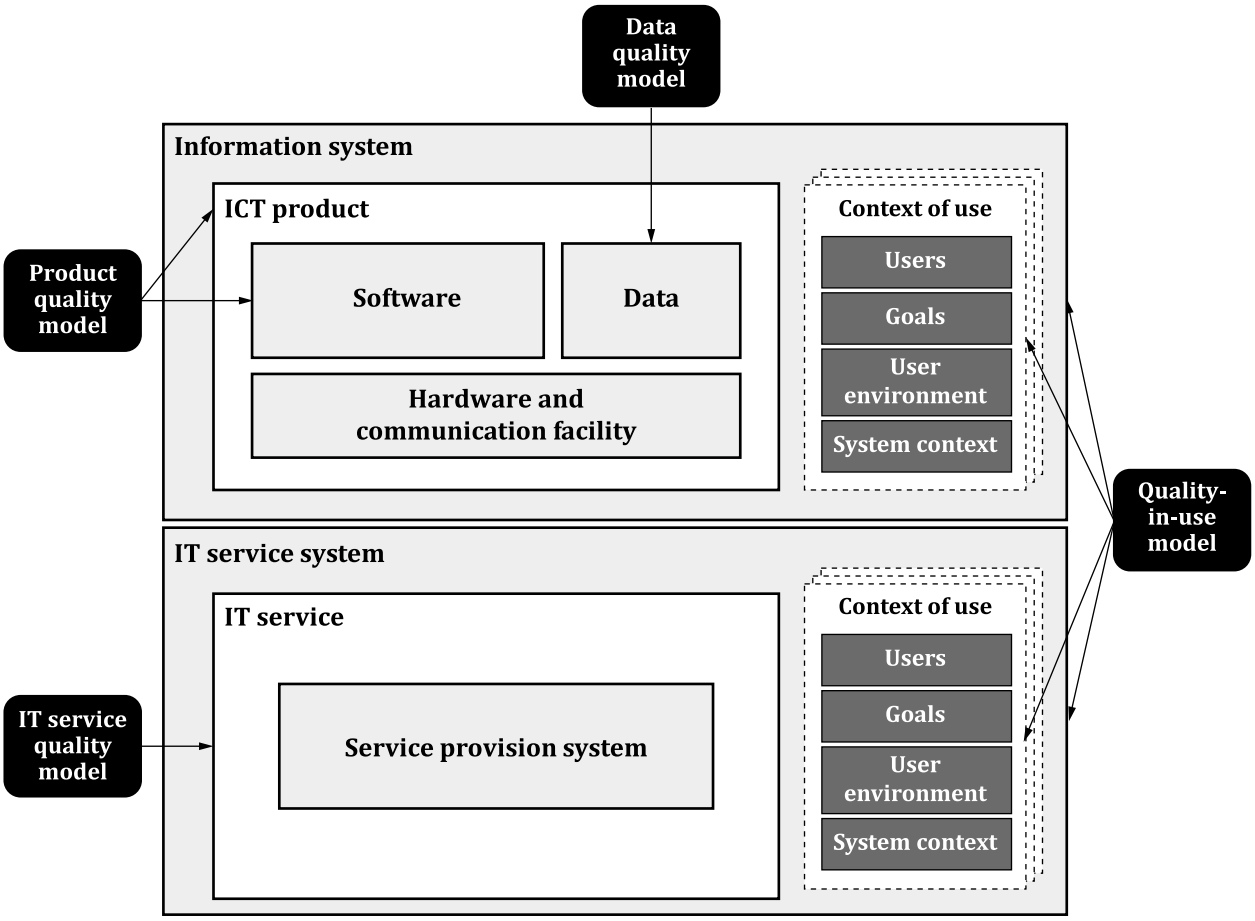
— Users: types of people who will be using the system

— Goals: users' intention to achieve by using the system

— User environment: situation in which users are using the system

— System context: nature and technical context of the system being used

The product quality model focuses on the target ICT product that includes the target software product. The target ICT product also includes hardware and communication facilities, as well as data which is the subject of the data quality model. The target ICT product is included in an information system that can also include one or more ICT products.



NOTE Arrows indicate the target entities within information systems or IT services to which 2501n quality models apply.

Figure 3 — Quality models and their target entities

NOTE 1 When an IT service does not contain an ICT product and its components, the IT service can be evaluated separately from them.

NOTE 2 The product quality model is indirectly applied to hardware and communication facilities when they are a component of an ICT product.

NOTE 3 Service provision system is an information system to provide IT service to users, including people, processes, technology, facilities and information.

5 Relationship to the quality-in-use model

The properties of the product determine the product quality in particular contexts of use (Table 1).

The functional suitability, performance efficiency, compatibility, interaction capability, reliability, security, flexibility and safety have a significant influence on the quality-in-use for primary users. Functional suitability, performance efficiency, interaction capability, reliability, security, flexibility and safety can also be specific concerns of other stakeholders who specialize in these areas.

Performance efficiency, compatibility, reliability, security, maintainability, flexibility and safety have a significant influence on quality-in-use for secondary users who maintain the system.

NOTE Types of users and stakeholders are explained in Annex D.

Table 1 — Influence of the quality characteristics

Product quality characteristics	Influence on quality-in-use for primary users	Influence on quality-in-use for maintenance tasks	Information system quality concerns of other stakeholders
Functional suitability	X		X
Performance efficiency	X	X	X
Compatibility	X	X	
Interaction capability	X		X
Reliability	X	X	X
Security	X	X	X
Maintainability		X	
Flexibility	X	X	X
Safety	X	X	X
Key X product quality influences quality-in-use for these stakeholders			

Annex A
(informative)

Comparison with the product quality model in ISO/IEC 25010:2011

This document revises ISO/IEC 25010:2011 and incorporates the same quality characteristics with some amendments.

— Safety has been added as a quality characteristic with subcharacteristics, i.e. operational constraint, risk identification, fail safe, hazard warning and safe integration.

— Usability and portability have been replaced with interaction capability and flexibility respectively.

— Inclusivity and self-descriptiveness, resistance, and scalability have been added as subcharacteristics of interaction capability, security, and flexibility respectively.

— User interface aesthetics and maturity have been replaced with user engagement and faultlessness respectively.

— Accessibility has been split into inclusivity and user assistance.

— Definitions have been based on those from existing ISO/IEC documents when possible; and terms defined in this document have been worded to represent the general meaning of the term.

— Several characteristics and subcharacteristics have been given more accurate names and definitions.

Table A.1 lists the differences of the characteristics and subcharacteristics between this document and ISO/IEC 25010:2011.

Table A.1 — Comparison with the previous product quality model in ISO/IEC 25010:2011

Clause	ISO/IEC 25010:2023	ISO/IEC 25010:2011	Notes
3.1	**Functional suitability**	**Functional suitability**	
3.1.1	Functional completeness	Functional completeness	
3.1.2	Functional correctness	Functional correctness	
3.1.3	Functional appropriateness	Functional appropriateness	
3.2	**Performance efficiency**	**Performance efficiency**	
3.2.1	Time behaviour	Time behaviour	
3.2.2	Resource utilization	Resource utilization	
3.2.3	Capacity	Capacity	
3.3	**Compatibility**	**Compatibility**	
3.3.1	Co-existence	Co-existence	
3.3.2	Interoperability	Interoperability	
3.4	**Interaction capability**	**Usability**	Renamed
3.4.1	Appropriateness recognizability	Appropriateness recognizability	
3.4.2	Learnability	Learnability	
3.4.3	Operability	Operability	
3.4.4	User error protection	User error protection	
3.4.5	User engagement	User interface aesthetics	Renamed

Table A.1 *(continued)*

Clause	ISO/IEC 25010:2023	ISO/IEC 25010:2011	Notes
3.4.6	Inclusivity	Accessibility	Split and renamed
3.4.7	User assistance		
3.4.8	Self-descriptiveness		New subcharacteristic
3.5	**Reliability**	**Reliability**	
3.5.1	Faultlessness	Maturity	Renamed
3.5.2	Availability	Availability	
3.5.3	Fault tolerance	Fault tolerance	
3.5.4	Recoverability	Recoverability	
3.6	**Security**	**Security**	
3.6.1	Confidentiality	Confidentiality	
3.6.2	Integrity	Integrity	
3.6.3	Non-repudiation	Non-repudiation	
3.6.4	Accountability	Accountability	
3.6.5	Authenticity	Authenticity	
3.6.6	Resistance		New subcharacteristic
3.7	**Maintainability**	**Maintainability**	
3.7.1	Modularity	Modularity	
3.7.2	Reusability	Reusability	
3.7.3	Analysability	Analysability	
3.7.4	Modifiability	Modifiability	
3.7.5	Testability	Testability	
3.8	**Flexibility**	**Portability**	Renamed
3.8.1	Adaptability	Adaptability	
3.8.2	Scalability		New subcharacteristic
3.8.3	Installability	Installability	
3.8.4	Replaceability	Replaceability	
3.9	**Safety**		New characteristic
3.9.1	Operational constraint		New subcharacteristic
3.9.2	Risk identification		New subcharacteristic
3.9.3	Fail safe		New subcharacteristic
3.9.4	Hazard warning		New subcharacteristic
3.9.5	Safe integration		New subcharacteristic

Annex B
(informative)

Example of mapping to dependability

This annex gives an example of how an organization can map its own model of software quality to the product quality model in this document.

Dependability is defined in IEC 60050-192 as the "ability to perform as and when required". One example of an alternative categorization of product qualities based on dependability[1] is:

— Availability: ability to be in a state to perform as required [SOURCE: IEC 60050-192-01-23]

— Reliability: ability to perform as required, without failure, for a given time interval, under given conditions [SOURCE: IEC 60050-192-01-24]

— Recoverability: ability to recover from a failure, without corrective maintenance [SOURCE: IEC 60050-192-01-25]

— Maintainability: ability to be retained in, or restored to a state to perform as required, under given conditions of use and maintenance [SOURCE: IEC 60050-192-01-27]

— Maintenance support performance: effectiveness of an organization in respect of maintenance support [SOURCE: IEC 60050-192-01-29]

— Durability: ability to perform as required, under given conditions of use and maintenance, until the end of useful life [SOURCE: IEC 60050-192-01-21]

— Safety: This term is not defined as vocabulary of dependability in IEC 60050-192.

— Security: This term is not defined as vocabulary of dependability in IEC 60050-192.

To conform to this document, this definition of dependability can be mapped onto the parts of the ISO/IEC 25010 quality model shown in Table B.1.

Table B.1 — Mapping of dependability

Clause	ISO/IEC 25010:2023	Dependability
3.1	**Functional suitability**	*
3.2	**Performance efficiency**	*
3.3	**Compatibility**	*
3.4	**Interaction capability**	*
3.5	**Reliability**	Reliability
3.5.2	Availability	Availability
3.5.3	Fault tolerance	Durability
3.5.4	Recoverability	Recoverability
3.6	**Security**	Security
3.7	**Maintainability**	Maintainability Maintenance support performance
3.8	**Flexibility**	*
3.9	**Safety**	Safety

If this definition of dependability was used as part of a wider assessment of product quality, it would also be necessary to consider functional suitability, performance efficiency, compatibility, interaction

capability, flexibility and quality-in-use characteristics and subcharacteristics, although these are not explicitly described to contribute to dependability defined in IEC 60050 (they are indicated by asterisk in Table B.1).

Annex C
(informative)

Using the quality model for measurement

C.1 Software properties

Some software properties are inherent in the software product; some are assigned to the software product. The quality of a software product in a particular context of use is determined by its inherent properties. Also, system properties can be often similar to software properties.

NOTE Examples of inherent properties are number of lines of code and the accuracy of a numeric calculation provided by the software. Examples of assigned properties are the owner of a software product, a warranty and the price of a software product.

Inherent properties can be classified as either functional properties or quality properties. Functional properties determine what the software is able to do. Quality properties determine how well the software performs. Quality properties are inherent to a software product and associated system. An assigned property is therefore not considered to be a quality characteristic of the software, since it can be changed without changing the software. Figure C.1 illustrates this classification of software properties.

Functional properties and quality properties are needed to be distinguished to be consistent to the basic concept that quality requirements are identified ones of non-functional requirements, distinguished from functional requirements, required to be achieved, evaluated and improved. Both of properties can support mutually. For examples, some of functional properties emerge sufficiently by supports from quality properties such as of faultless, fault tolerant, enough real time processing etc. Conversely, quality properties can be fleshed out with functional properties.

<table>
<tr><td rowspan="3">Software properties</td><td rowspan="2">Inherent properties</td><td>Domain-specific functional properties</td></tr>
<tr><td>Quality properties (functional suitability, reliability, performance efficiency, interaction capability, security, compatibility, maintainability, flexibility, safety)</td></tr>
<tr><td>Assigned properties</td><td>Business properties like, for example, price, delivery date, future product, product supplier</td></tr>
</table>

Figure C.1 — Software properties

C.2 Quality measures on internal properties, quality measures on external properties and quality-in-use measures

For each subcharacteristic, the capability of the software is determined by a set of static internal properties that can be measured. Examples of quality measures on internal properties (i.e. internal measures) are given in ISO/IEC 25023 and ISO/IEC 5055. The characteristics and subcharacteristics can be measured externally by the extent to which the capability is provided by the system containing the software. Examples of quality measures on external properties (i.e. external measures) are also given in ISO/IEC 25023.

Quality measures on external properties of system/software quality provide a “black box” view of the system/software and address properties related to the execution of the software on computer hardware and an operating system. Quality measures on internal properties of software quality provide a “white box” view of software and address static properties of the software product that are typically available to be evaluated during the development. The quality of software measured internally has an impact on the quality of system/software measured externally, which again has an impact on the quality-in-use of the system.

EXAMPLE Operability measured internally by the degree of conformance to menu interface design guidelines in ISO 9241-14 contributes to operability measured externally by the extent to which users can successfully manipulate the menus, which contributes to the effectiveness, efficiency and satisfaction when carrying out tasks (usability in the quality-in-use model).

Quality measures on internal properties based on inspecting static properties can be used to measure inherent properties of a software work product (see Table C.1). Static analysis methods include inspection and automated analysis tools. Work products include requirements and design documents, source code, and test procedures.

Quality measures on external properties of dynamic properties can be used to measure inherent properties of a computer system (the target computer system in Figure 3), and system-dependent properties of a software product.

Quality-in-use measures (derived from testing or observing the results of real or simulated use) measure intrinsic properties of a system that can include software, hardware, communications and users, and system dependent properties of a software-intensive computer system or of a software product. Quality-in-use measures relate to the impact of the system on stakeholders.

Table C.1 — Differences between quality measures on internal properties, quality measures on external properties and quality-in-use measures

Type of properties measured	**Software product properties**	**Computer system behaviour properties**	**Human-computer system impact properties**
Type of quality measure	Internal: inspection of static properties	External: test or modelling of dynamic properties	Quality-in-use: test or observation of results of real or simulated use
Type of properties of software product	Inherent	Computer system-dependent	Human-computer system-dependent
Type of properties of computer system		Inherent	Human-computer system-dependent
Type of properties of human-computer system			Inherent

Quality measures on internal properties of software can be used at an early stage of the system/software development process to predict quality measures on external properties of system/software quality. There are often quality measures on internal properties and quality measures on external properties for the same property, for example, a quality measure on internal property to estimate the expected response time to predict the time measured externally.

Examples of product quality measures are given in ISO/IEC 25023.

Annex D
(informative)

Quality from different stakeholders' perspectives

The quality models provide a framework for collecting stakeholder needs. Stakeholders include the following types of user and other types of people;

a) Primary users: person who interacts with the system to achieve the primary goals.

b) Secondary users: who provide support, for example

   1) content provider, system manager/administrator, security manager;

   2) maintainer, analyser, porter, installer.

c) Indirect users: person who receives output, but does not interact with the system.

d) Other stakeholders:

   1) those who are concerned with the product by business; examples include product salesman, executives of the product marketing company;

   2) those who are concerned with the product by engineering for its lifecycle; examples include requirements analysts, developer;

   3) public under the effects or influence of the product during its operation; examples include passengers of railway, users of electric power supply

Each of these types of user has needs for quality-in-use and product quality in particular contexts of use, as illustrated for some examples of users and quality characteristics by the questions shown in Table D.1.

NOTE The content provider also has user needs for data quality.

Table D.1 — Examples of user needs for product quality

User needs	Primary user	Secondary users		Indirect user
		Content provider	Maintainer	
	Interacting	Interacting	Maintaining or porting	Using output
Reliability	How reliable does the system need to be when the user uses it to perform their task?	How reliable does updating the system with new content need to be?	How reliable does maintaining or porting the system need to be?	How reliable does the output from the system need to be?
Security	How secure does the system need to be when the user uses it to perform their task?	How secure does the system need to be after the content provider updates it?	How secure does the system need to be after maintenance changes are made or when it is ported?	How secure does the output from the system need to be?
Learnability	To what extent does learning to use the system need to be effective, efficient, risk free and satisfying?	To what extent does learning to provide content need to be effective, efficient, risk free and satisfying?	To what extent does learning to maintain or port the system need to be effective, efficient, risk free and satisfying?	To what extent does learning to use the output from the system need to be effective, efficient, risk free and satisfying?
Inclusivity and user assistance	To what extent does the system need to be effective, efficient, risk free and satisfying to use for people with disabilities?	To what extent does providing content for the system need to be effective, efficient, risk free and satisfying for people with disabilities?	To what extent does maintaining or porting the system need to be effective, efficient, risk free and satisfying for people with disabilities?	To what extent does using the output of the system need to be effective, efficient, risk free and satisfying for people with disabilities?

The user needs in Table D.1 provide examples of starting points for requirements, and can be used as a basis for measuring the impact of the quality of the system on use and maintenance.

Prior to software development or acquisition, quality requirements should be defined from the perspective of stakeholders. Analysis of the quality-in-use requirements results in derived functional and quality requirements needed for a product to achieve the quality-in-use requirements.

EXAMPLE Overall needs for system reliability can lead to specific requirements for software product maturity, availability, fault tolerance and recoverability. Reliability can also have an impact on overall system effectiveness, efficiency, freedom from risk and satisfaction.

Bibliography

[1] IEC 60050-192:2015, *International Electrotechnical Vocabulary (IEV) — Part 192: Dependability*

[2] IEC 60050-192:2015/AMD1:2016, *Amendment 1 — International Electrotechnical Vocabulary (IEV) — Part 192: Dependability*

[3] IEEE 1517:2010, *IEEE Standard for Information Technology — Software Life Cycle Processes — Reuse Processes*

[4] ISO 7498-2:1989, *Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 2: Security Architecture*

[5] ISO 9001, *Quality management systems — Requirements*

[6] ISO/IEC 25023, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality*

[7] ISO 9241-14, *Ergonomic requirements for office work with visual display terminals (VDTs) — Part 14: Menu dialogues*

[8] ISO 9241-110, *Ergonomics of human-system interaction — Part 110: Interaction principles*

[9] ISO 9241-112, *Ergonomics of human-system interaction — Part 112: Principles for the presentation of information*

[10] ISO/IEC/IEEE 12207:2017, *Systems and software engineering — Software life cycle processes*

[11] ISO/IEC/IEEE 15288, *Systems and software engineering — System life cycle processes*

[12] ISO/IEC/IEEE 24765:2017, *Systems and software engineering — Vocabulary*

[13] ISO/IEC 25000, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE*

[14] ISO/IEC 25002, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality models overview and usage*

[15] ISO/IEC TS 25011, *Information technology — Systems and software Quality Requirements and Evaluation (SQuaRE) — Service quality models*

[16] ISO/IEC 25012, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Data quality model*

[17] ISO/IEC 25019, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality-in-use model*

[18] ISO/IEC 25020, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality measurement framework*

[19] ISO/IEC 25030, *Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — Quality requirements framework*

[20] ISO/IEC 25040, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation process*

[21] ISO/IEC 25021, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality measure elements*

[22] ISO/IEC 33063, *Information technology — Process assessment — Process assessment model for software testing*

[23] ISO/IEC/IEEE 29119-1, *Software and systems engineering — Software testing — Part 1: General concepts*

[24] ISO/IEC 5055, *Information technology — Software measurement — Software quality measurement — Automated source code quality measures*

[25] ISO TR 25060, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — General framework for Common Industry Format (CIF) for usability-related information*

ICS 35.080

Price based on 22 pages