INTERNATIONAL STANDARD

ISO/IEC 25030

Second edition
2019-08

Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — Quality requirements framework

Reference number
ISO/IEC 25030:2019(E)

COPYRIGHT PROTECTED DOCUMENT

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

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).

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 http://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.

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

This second edition cancels and replaces the first edition (ISO/IEC 25030:2007), which has been technically revised.

The main changes compared to the previous edition are as follows:

— extension of the view from software to system;

— enhancement and deployment of quality requirements;

— clarification of quality requirements definition steps:

  — stating them exhaustively by using the quality models;

  — specifying them with the quality measures with criteria for evaluation;

— clarification of how to use quality requirements.

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.

Introduction

It is important to identify and specify quality requirements as part of system, software and data requirements, because finding the right balance of quality requirements, in addition to well-specified functional requirements, is a critical success factor to meet the stakeholders' objectives. Quality requirements are needed for:

— specifying the system, including contractual agreements and call for tender;

— planning the project, including feasibility analysis;

— developing the system, including identification of architecture drivers or potential quality problems during development; and

— evaluating the system, including objective assessment and certification of quality.

This document focuses on defining, using and governing quality requirements. If not clearly defined, they can be viewed, interpreted, implemented and evaluated differently by the relevant stakeholders. This can result in systems that are inconsistent with user expectations and of poor quality; and time and cost overruns to rework the system. Therefore quality requirements for the system need to be specified clearly at the earliest stage of the development or acquiring process as possible, to provide a critical input to the development or acquisition.

This document can be used to improve the quality of quality requirements, by providing requirements and recommendations for them, and provides guidance for the steps used to define and use them.

Quality requirements can be categorized into characteristics/subcharacteristics by using the quality models defined in the ISO/IEC 2501n family of standards. Measures of these characteristics/subcharacteristics, which are defined in the ISO/IEC 2502n family of standards, can be used to specify quality requirements and evaluate the quality of the target system or data. After ISO/IEC 25030:2007 was published, several international standards which define these models and measures have been published and so the previous edition has become inconsistent with these standards.

Furthermore many systems are now deeply embedded into social infrastructures used in daily life. This requires the systems to achieve much higher quality; e.g., connected systems need to be interoperable and secure, reliable, maintainable and usable.

This revision updates the quality requirements division of SQuaRE series, aligning it with the other divisions, and furthermore providing more practical guidelines for defining and using quality requirements.

Figure 1 illustrates the organization of the SQuaRE series representing families of standards, further called divisions. The SQuaRE series consists of five main divisions and on extension division. The divisions within the SQuaRE series are:

— **ISO/IEC 2500n — Quality Management Division**. The standards that form this division define all common models, terms and definitions used by all other standards in the SQuaRE series. The division also provides requirements and guidance for the planning and management of a project.

— **ISO/IEC 2501n — Quality Model Division**. The standards that form this division provide quality models for system/software products, quality in use (QIU), data, and IT services. Practical guidance on the use of the quality model is also provided.

— **ISO/IEC 2502n — Quality Measurement Division**. The standards that form this division include a system/software product quality measurement reference model, definitions of quality measures, and practical guidance for their application. This division presents internal measures of software quality, external measures of software quality, QIU measures and data quality measures. Quality measure elements forming foundations for the quality measures are defined and presented.

— **ISO/IEC 2503n — Quality Requirements Division**. The standard that forms this division helps specifying quality requirements. These quality requirements can be used in the process of quality

requirements elicitation for a system/software product to be developed, designing a process for achieving necessary quality, or as inputs for an evaluation process.

— **ISO/IEC 2504n — Quality Evaluation Division**. The standards that form this division provide requirements, recommendations and guidelines for system/software product evaluation, whether performed by independent evaluators, acquirers or developers. The support for documenting a measure as an Evaluation Module is also presented.

ISO/IEC 25050 to ISO/IEC 25099 are reserved for SQuaRE extension International Standards, which currently include in ISO/IEC 25051 requirements for quality of Ready to Use Software Products (RUSP) and instructions for testing, and in ISO/IEC TR 25060 to ISO/IEC 25069 common industry format for usability.

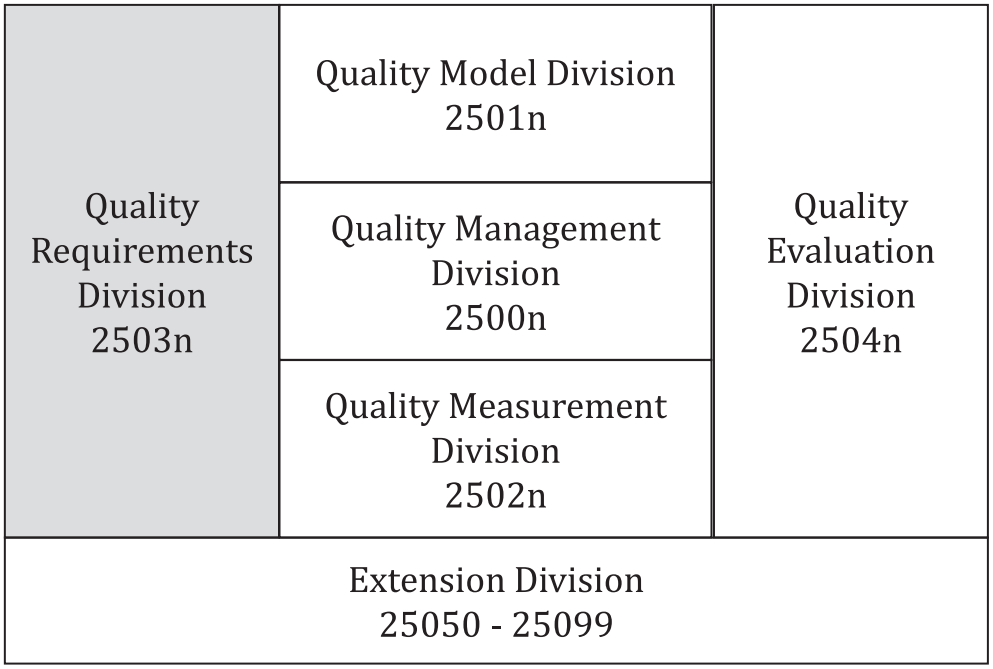

Figure 1 — Organization of the SQuaRE series of International Standards

Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — Quality requirements framework

1 Scope

This document provides the framework for quality requirements for systems, software products and data, which includes concept of the quality requirements, and requirements and recommendations for the processes and methods to elicit, define, use and govern them. Intended readers of this document include, but are not limited to:

— acquirers: evaluate if the system/software products/data fulfills their value proposition, i.e., meets the expected quality,

— developers: design, implement and test the system/software products/data to ensure that it meets the expected quality,

— testers: verify and validate that the system/software products/data meets the expected quality,

— project managers: plan, monitor and control the achievement of the expected quality, and

— independent evaluators: evaluate the system/software products/data with the objective criteria.

This document complies with the technical processes defined in ISO/IEC/IEEE 15288, which are relevant for elicitation of stakeholders' quality needs and for defining, analyzing and maintaining quality requirements. In this document, the quality models in ISO/IEC 25010 and ISO/IEC 25012 are used to categorize quality requirements and to provide a basis for quantifying them in terms of quality measures in the quality measure division of ISO/IEC 2502n.

This document does not cover specification of the other requirements (such as functional requirements, process requirements, etc.), and prescribes neither any specific quality measure nor any specific development process.

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.

ISO/IEC 25000:2014, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE*

ISO/IEC 25010:2011, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*

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

ISO/IEC 25022, *Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — Measurement of quality in use*

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

ISO/IEC 25024, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of data quality*

ISO/IEC/IEEE 15288:2015, *Systems and software engineering — System life cycle processes*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 25000 and the following apply.

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

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

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

NOTE The essential definitions from ISO/IEC 25000 and the other ISO standards are reproduced here.

3.1
classification axis
total range of a mapping of systems and software for categorizing them from a particular perspective

[SOURCE: ISO/IEC TR 12182:2015, 3.7]

3.2
context of use
conditions and constraints under which *ICT products* (3.8) are used by specific *users* (3.20) in a specific environment to achieve specific goals as part of the larger *information system* (3.10)

Note 1 to entry: Environment includes physical aspects such as equipment and resources as well as social aspects such as demographics and culture.

3.3
deployment
deployment of requirements
assignment of *requirements* (3.16) along with the system decomposition

3.4
derivation
derivation of requirements
translation and elaboration of *requirements* (3.16) from one type of requirements to another in the same system level

Note 1 to entry: Types of requirements include *quality in use* (3.13) requirements, product *quality requirements* (3.15) and data requirements.

3.5
domain-based requirement
requirement (3.16) originated from its application domain

3.6
functional requirement
requirement (3.16) that specifies a function that a system or system component shall perform

[SOURCE: IEEE 730:2014, 3.2]

3.7
ICT requirement
requirement (3.16) resulting from adoption of some information and communication technologies (ICTs) technical solutions in the design process

Note 1 to entry: ICT technical solutions include web-based technologies, cloud servers, and so on.

3.8
ICT product
product (3.12) which uses information and communication technologies (ICTs) and can be a part of *information system* (3.10)

Note 1 to entry: Figure 3 describes what ICT product consists of and the relationship to information system.

3.9
indirect user
person who receives output from a system, but does not interact with the system

EXAMPLE Executive manager, service acquirer.

[SOURCE: ISO/IEC 25010:2011, 4.3.6, modified — EXAMPLE has been added.]

3.10
information system
system that comprises of software, hardware, communication facility, data and the people who use it in a given environment to satisfy their information processing needs

Note 1 to entry: Figure 3 describes what information system consists of.

3.11
primary user
user (3.20) who interacts with the system to achieve the primary goals

Note 1 to entry: The definition is adapted from ISO/IEC 25010:2011, 3.6.

3.12
product
artifact that is produced, is quantifiable and is deliverable to the *user* (3.20) as either an end item in itself or a component item

Note 1 to entry: This definition is adapted from A Guide to the Project Management Body of Knowledge (PMBOK) Fifth Edition.

Note 2 to entry: Product includes *ICT products* (3.8), software, and software components.

3.13
quality in use
extent to which the behavioral and attitudinal outcomes and consequences of use of a *product* (3.12), system or service meets the needs of *users* (3.20) or other *stakeholders* (3.18) in specific *contexts of use* (3.2)

3.14
quality measure
measure that is defined as a measurement function of two or more values of quality measure elements

[SOURCE: ISO/IEC 25010:2011, 4.3.10]

3.15
quality requirement
requirement (3.16) for quality properties or attributes of an *ICT product* (3.8), data or service that satisfy needs which ensue from the purpose for which that ICT product, data or service is to be used

Note 1 to entry: Quality requirements in this document do not cover quality requirements for service.

3.16
requirement
statement which translates or expresses a need and its associated constraints and conditions

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.37]

3.17
secondary user
user (3.20) who interacts with the *product* (3.12) to support the *primary users* (3.11)

EXAMPLE Content provider, system manager, administrator, security manager, maintainer, installer.

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.3651, modified — The word "person" has been replaced with "user"; EXAMPLE has been added.]

3.18
stakeholder
individual or organization having a right, share, claim or interest in a system or in its possession of characteristics that meet their needs and expectations

Note 1 to entry: Stakeholders include *users* (3.20), developers, testers, project managers, acquirers, independent evaluators, data owners, supporters, trainers, regulatory bodies and other people influenced by the system.

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.44, modified — The original EXAMPLE and Note 1 to entry have been replaced with a new Note 1 to entry.]

3.19
technical product quality requirement
product (3.12) *quality requirement* (3.15) on its technically identified properties which are used in its development and maintenance processes

3.20
user
individual or group that interacts with a system or benefits from a system during its utilization

[SOURCE: ISO/IEC 25010:2011, 4.3.16, modified — NOTE has been removed.]

3.21
validation
confirmation, through the provision of objective evidence, that the *requirements* (3.16) for a specific intended use or application have been fulfilled

[SOURCE: ISO/IEC 25000:2014, 4.41, modified — Note 1 to entry has been removed.]

3.22
verification
confirmation, through the provision of objective evidence, that specified *requirements* (3.16) have been fulfilled

[SOURCE: ISO/IEC 25000:2014, 4.43, modified — Note 1 to entry has been removed.]

4 Abbreviated terms

ICT	information and communication technology
PQR	product quality requirement
QIUR	quality in use requirement
DQR	data quality requirement
SRS	software requirements specification
StRS	stakeholder requirements specification
SyRS	system requirements specification

5 Conformance

Any quality requirements specification that conforms to this document shall meet all the requirements described in Clauses 6, 7 and 8.

6 Concept of quality requirements

6.1 General

This clause describes the concept of quality requirements, including their target entities for which the quality requirements are to be defined, and important considerations on them.

6.2 Types of quality requirements

Quality in use requirements (QIURs) specify the required levels of quality from the stakeholders' point of view. These requirements are derived from the needs of various stakeholders. QIURs relate to the outcome when the product is used in a particular context of use, and QIURs can be used as the target for validation of the product.

Product quality requirements (PQRs) specify levels of quality required from the viewpoint of the ICT product. Most of them are derived from stakeholder quality requirements including QIURs, which can be used as targets for verification and validation of the target ICT product. Technical product quality requirements are requirements for technically identified attributes (targeting specifications, source code, etc.) to meet the other PQRs. Technical product quality requirements can be used as targets for verification at various stages of development and maintenance.

NOTE 1 PQRs can also be used to specify attributes of deliverable, non-executable software products such as documentation and manuals.

The data quality requirements (DQRs) specify levels of quality required for the data associated with the product. These include requirements derived from QIURs and PQRs of input and output products. DQRs can be used for verification and validation from the data side.

NOTE 2 Many DQRs can be derived from PQRs for the target product, while some DQRs such as data integrity can be derived directly from QIURs.

6.3 Targets for quality requirements

The scope of the three types of quality requirements is shown in Figure 2. QIURs are defined on the information system, which includes not only an ICT product but also its users and relevant environments (e.g., mechanicals monitored/controlled by the ICT product and business processes in which the ICT product is used). PQRs are defined on the ICT product or its constituents (including sub-ICT products, hardware, communication facilities, software and, in some cases, software components), and DQRs are defined on the data inside the ICT product.

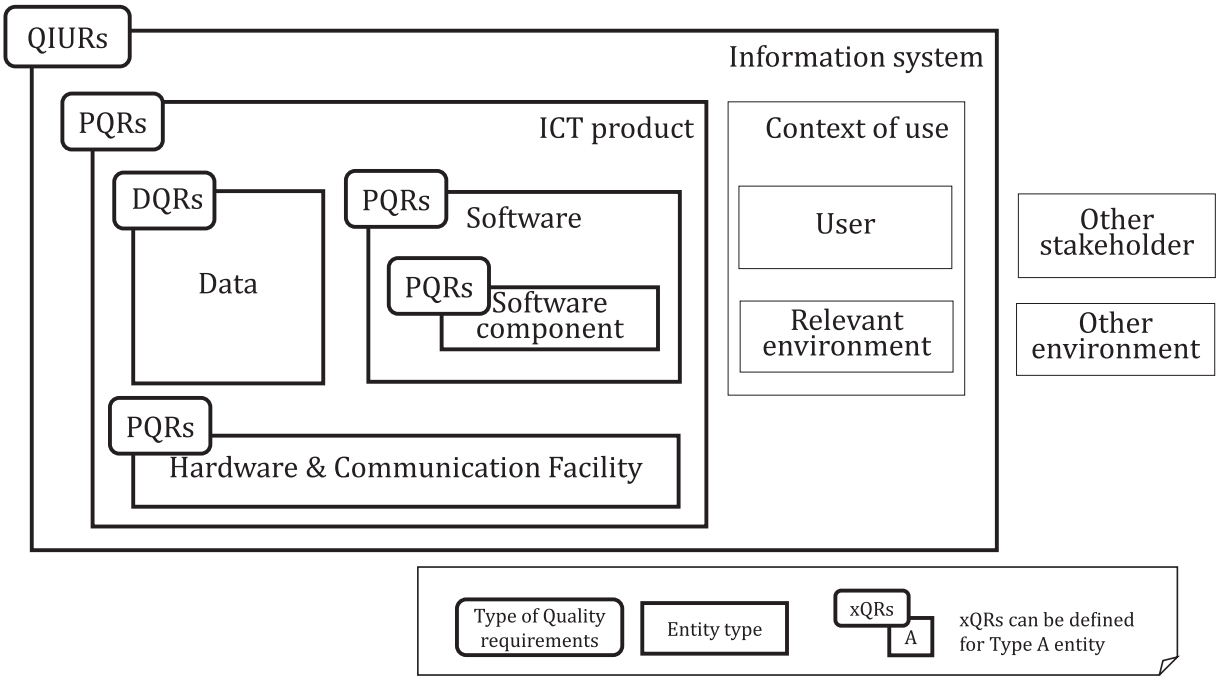

Figure 2 — Scope of quality requirements

Figure 2 describes only the scope of each type of quality requirements, not describing the system hierarchy, which is formally defined in Figure 3.

NOTE 1 Annex K describes how IT service quality requirements are to be treated.

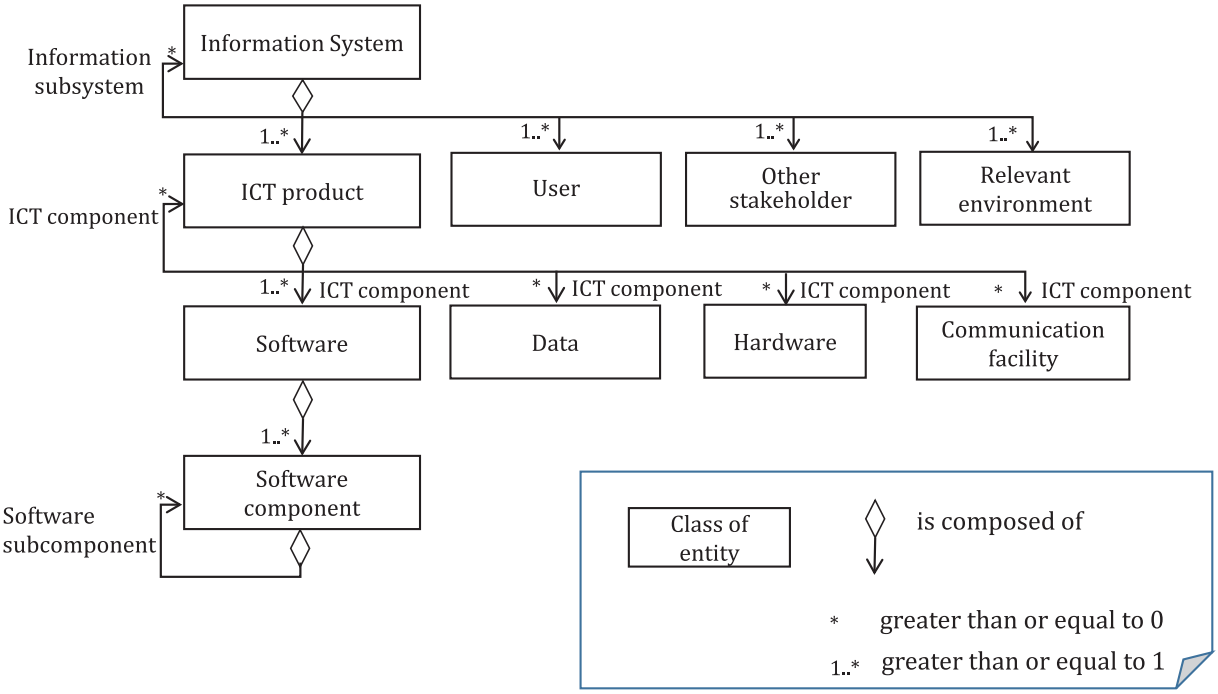

Figure 3 — System hierarchy used in Figure 2

NOTE 2 Users include primary users, secondary users and indirect users. See Table 2.

NOTE 3 A “system of systems” can be considered an information system, which recursively includes some subsidiary information systems.

NOTE 4 An ICT product includes software, and also can include data, hardware, communication facilities, and other ICT products as its ICT components.

6.4 Quality models and measures for quality requirements

Quality requirements are defined by using quality models and quality measures. Table 1 shows which International Standards can be used for defining each type of quality requirements.

Table 1 — Quality models and measures for quality requirements

Quality requirements	Quality model	Quality measure
QIURs	ISO/IEC 25010 Quality in use model	ISO/IEC 25022 Measurement of quality in use
PQRs	ISO/IEC 25010 System and software product quality model	ISO/IEC 25023 Measurement of system and software product quality
DQRs	ISO/IEC 25012 Data quality model	ISO/IEC 25024 Measurement of data quality

ISO/IEC 25022, ISO/IEC 25023 and ISO/IEC 25024 provide a list of quality measures in a tabular form, categorised by quality characteristics and subcharacteristics. The following information is given for each quality measure in the tables.

ID: Identification code of the quality measure.

Name: Quality measure name.

Description: Information provided by the quality measure.

Measurement function: Mathematical formula showing how the quality measure elements are combined to produce the quality measure.

NOTE Each quality measure listed in ISO/IEC 25022 can be used to measure effectiveness, efficiency, satisfaction and freedom from risk in specific contexts of use. Each quality measure listed in ISO/IEC 25023 can be used to measure internal properties (typically static measures of intermediate products), external properties (typically by measuring the behavior of the code when executed) or both. Each quality measure listed in ISO/IEC 25024 can be used to measure inherent or system dependent properties.

6.5 Important considerations of quality requirements

6.5.1 Sources of quality requirements

Two types of requirements for ICT products should be considered based on their sources: domain-based requirements, which are derived directly from stakeholder needs for their domain through requirements analysis processes, and ICT requirements, which are newly introduced by the adoption of some ICT technical solutions through design processes. Quality requirements also have the same types.

An example of ICT requirements is as follows. Adopting a web-based system (ICT technical solution) entails some user requirements like how to behave when clicking the back button on browsers (functional requirement), and self-descriptiveness of its user interface (PQR: learnability), and browser compatibility (PQR).

6.5.2 Categories of ICT products

Quality required for one ICT product is different from that for another, and therefore the category of the target system is crucially important to determine which quality characteristics have higher priority and which quality measures should be used.

ISO/IEC TR 12182 provides the framework for categorizations of ICT products, including an exemplary set of classification axes, which are organized hierarchically with the four axes in the first layer: Architecture/Structure, Property, Operational environment, Data, and Stakeholder of target system. These axes can be used for determining which quality characteristics have higher priority. Classification axes which are important for determining priority can include:

— Function (and its problem frame)

— Criticality of system and data

— Stakeholders' characteristics

NOTE Examples of supporting IT decisions on required level of quality are shown in Annex J.

6.5.3 Interrelation with functional/data requirements

Quality requirements cannot be defined and analyzed separately from functional/data requirements. Some quality requirements are attached to functional/data requirements; and also some quality requirements are achieved by specifying requirements for new functions.

EXAMPLE 1 Quality requirements that are attached to functional requirements:

— Time efficiency (response time) is defined as response time for a function

EXAMPLE 2 Quality requirements that are achieved by specifying requirements for new functions:

— Some confidentiality requirements are achieved by requirements for access control function

— Some learnability requirements are achieved by requirements for help function

— Some analyzability requirements are achieved by requirements for logging function

NOTE 1 Unlike functional requirements, most quality requirements represent emergent properties of the system, which appear on a set of components, not on a specific component. Thus it is hard to build and maintain the traceability of quality requirements and as a result to surely implement and verify them throughout the product lifecycle.

Product requirements cannot be defined separately from the data requirements. ICT products consume and generate data. Quality requirements (product quality requirements or data quality requirements) can be developed to product quality requirements and/or data quality requirements along with system decomposition.

NOTE 2 There are three types of data from the perspective of ICT product: input and/or output data outside, data which its components exchange and configuration data.

NOTE 3 An example of the interdependence between the ICT product and data, and the relationship between their PQRs and DQRs is shown below:

— The configuration data file is written for the configuring the ICT product. Its DQRs (e.g., flexibility requirements) are determined from the functions and quality requirements to be fulfilled by ICT products.

— The quality (e.g., accuracy) of customer data, which is an input to a business support system, affects the product quality (e.g., interoperability and functional correctness) of the system.

— Data that are exchanged among software components of the ICT product, and their DQRs (e.g., efficiency) strongly influence the method of component implementation and product quality (e.g., time efficiency) of the ICT product.

6.5.4 Derivation of quality requirements

In the case of a large-scale system, it is important to grasp where the target entity of the quality requirements is located in the system hierarchy because the quality requirements are often derived from the higher level of entities to the lower levels.

Figure 4 describes how each type of quality requirements for entities at some level derive others.

The primary source of quality requirements is users, from whom first QIURs for the information system including the target entities are elicited and documented. Then they evolve into PQRs and DQRs for the target entities. Other stakeholders, such as developers and regulatory bodies, also give some quality requirements on the target entities. Finally other entities give some requirements as constraints to the target entities, including non-target ICT products, software and data which are connected to or used in the targets, and hardware and communication which are used in them.

NOTE A QIUR can derive a set of functions to each of which some PQRs are attached, e.g., an efficiency requirement of a user task can derive a function to automate some portion of the task with a time behavior requirement. Similarly in their deployment to the lower level of targets, PQRs can also derive a set of functions to each of which some PQRs are attached. See 6.5.3.

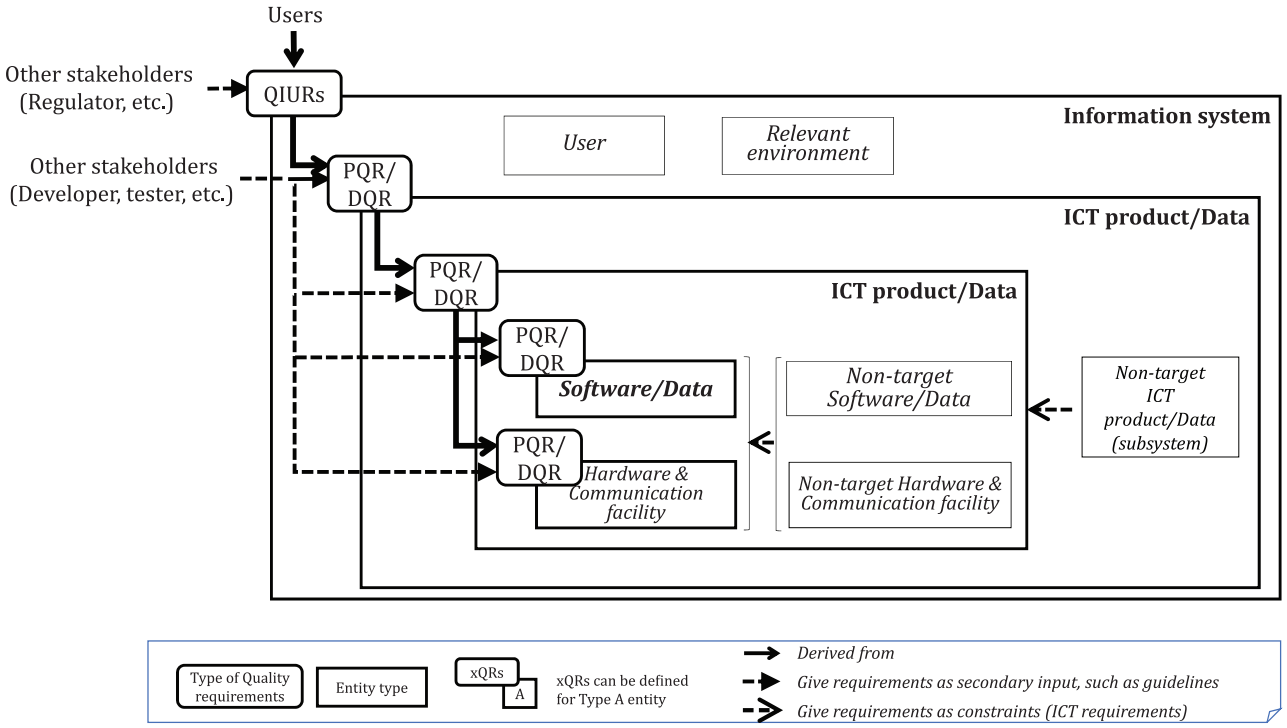

Figure 4 — Derivation of quality requirements

6.5.5 Quality requirements trade-offs

Quality requirements can have conflicts with one other. These conflicts inevitably arise as a result of the inter-relationships between quality characteristics. In case that some quality characteristic can negatively influence on another, trade-offs should be conducted to resolve the conflict to find the right balance of them.

NOTE Examples of inter-relationship between product quality characteristics are shown in Annex G.

7 Quality requirements processes

7.1 General

This clause describes the requirements and recommendations for quality requirements processes on how quality requirements are prepared, defined and analyzed.

7.2 Overview of quality requirements processes

The quality requirements shall be elicited, defined, analyzed and maintained by using the requirements-related processes defined in ISO/IEC/IEEE 15288: stakeholder needs and requirements definition process and system requirements definition process, through which the stakeholder needs are to be elicited and transformed into the system requirements. See Figure 5.

NOTE 1 Both a system and a software product are regarded as an ICT product in this document.

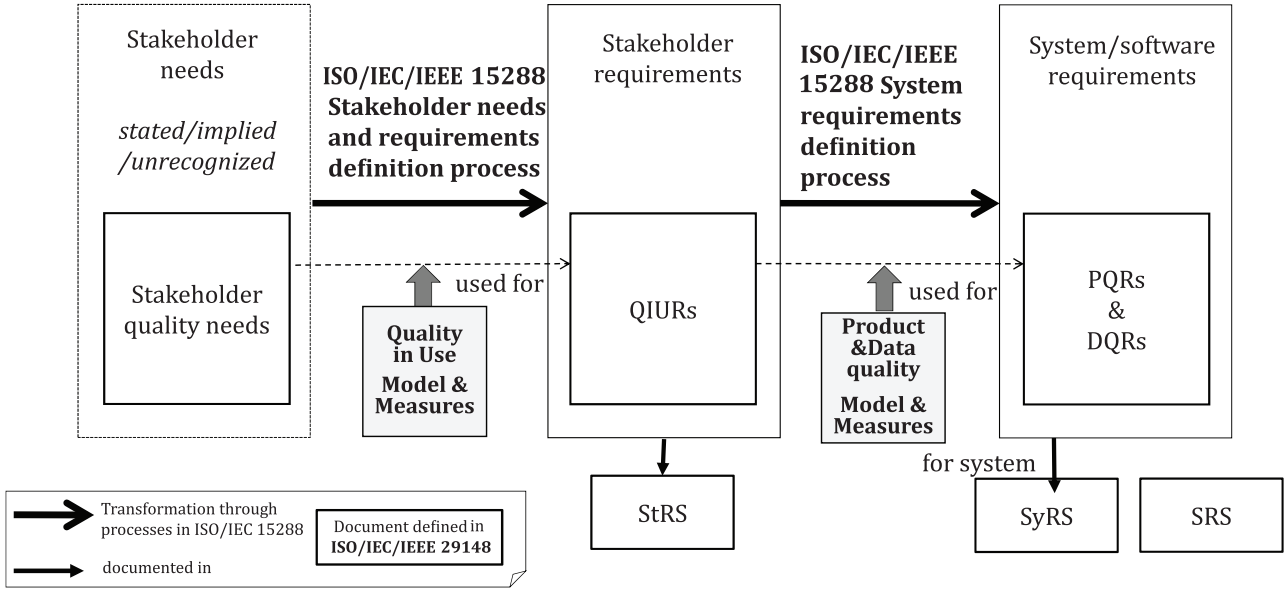

Figure 5 — From stakeholder needs to system/software requirements

The stakeholder needs and requirements definition process identifies stakeholders, or stakeholder classes, involved with the system throughout its life cycle, and their needs. It analyzes and transforms these needs into a common set of stakeholder requirements that express the intended interaction the system will have with its operational environment and that is the reference against which each resulting operational capability is validated. Quality needs for the target system as part of stakeholder needs are also elicited and transformed into QUIRs as part of stakeholder requirements, using the quality in use model and measures.

The system requirements definition process creates a set of measurable system requirements that specify, from the supplier's perspective, what characteristics, attributes, and functional and performance requirements the system is to possess, in order to satisfy stakeholder requirements. PQRs and DQRs as part of system requirements are also defined and analyzed in order to meet stakeholder requirements, using the product and data quality model and measures.

NOTE 2 The detailed relationship to ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 29148 is respectively described in Annex D and Annex E. ISO/IEC/IEEE 29148 associates the above two requirements related processes with the processes in ISO/IEC/IEEE 12207, which is another International Standard that defines the software lifecycle processes.

NOTE 3 Iteration and recursion in requirements engineering are described in ISO/IEC/IEEE 29148. The requirements, architecture and design processes can be iteratively applied on the same level of the system, in order to resolve trade-offs between the requirements and architecture. These set of processes can also be recursively applied successive levels of system elements within the system structure for successful engineering of systems beyond trivial complexity.

NOTE 4 In general quality requirements are more stable than functional requirements over the product lifecycle; however they can also change, e.g. security requirements need to be changed if new functionality is added, and interoperability requirements need to be reconsidered if the environment changes even a little.

NOTE 5 As defined in ISO/IEC/IEEE 29148, the stakeholder requirements can be documented in the stakeholder requirements specification (StRS), and the system and software requirements can be documented respectively in the system requirements specification (SyRS) and the software requirements specification (SRS).

7.3 Elicitation of quality needs

7.3.1 Identification of stakeholders

This subclause relates to the identification of stakeholders in the activities of "(a) Prepare for stakeholder needs and requirements definition" for Stakeholder needs and requirements definition process in ISO/IEC/IEEE 15288 (See Annex D).

The representatives from all groups of the stakeholders who are a potential source of the quality requirements shall be identified and, if available, involved for eliciting them. Table 2 describes which type of stakeholders is a source of, a user of and relevant to which type of quality requirements.

Table 2 — Stakeholders and types of quality requirements

Quality requirement	Stakeholder					
	User			Other stakeholder		
	Primary user	Secondary user	Indirect user	Developer	Acquirer/ Independent evaluator	Society
QIUR	S	S	S	U	U	R
PQR	S	S		U	U	
PQR Technical				S, U	U	
DQR	S	S	S	S, U	U	

Key

S: a source of

U: a user of

R: relevant to

The users of quality requirements (developer, acquirer/independent evaluator etc.) have a responsibility for establishing and maintaining the quality requirements, and so they should consider freedom of risks requirements relevant to society, who is a group of persons (who are supposed to be) influenced by the system, who cannot directly be a source of requirements.

NOTE Stakeholder can be considered as a role, and so one person or organization can have more than one role. Moreover a stakeholder is not supposed to belong to a specific kind of organization. For example, in the case of developing consumer products, acquirers and developers can belong to the same company to consider any risks on people influenced by the system.

7.3.2 Defining stakeholder needs

This subclause relates to the activities of "(b) Define stakeholder needs" for Stakeholder needs and requirements definition process in ISO/IEC/IEEE 15288 (See Annex D).

The assumed context of use for the target information system and the quality needs in the context of use shall be extracted from the identified stakeholders. If there is an existing system, they shall also be extracted from analyzing feedback on the experience with using the system from the stakeholders.

NOTE 1 Annex A provides a recommended process for elicitation of quality needs.

NOTE 2 Quality needs in this case mainly relate to quality in use.

NOTE 3 Stakeholder needs include not only those stated explicitly but also those implicit or unrecognized. To elicit relevant stakeholder needs in a particular context of use exhaustively, the stakeholder-target matrix can be used, shown in Annex I.

The elicited stakeholders' quality needs shall be prioritized and be selected based on it, where the categories of the ICT product (6.5.2) should be taken into account to determine which quality characteristics are important to the target entity.

NOTE 4 Since different stakeholders have different needs for the target system, the stakeholder needs can be inconsistent and/or incomplete; therefore all the prepared needs are examined to define, analyze and maintain a set of stakeholders' requirements.

NOTE 5 Not all stakeholder needs can be selected to define stakeholder requirements due to some business reasons. For example, a package software provider can decide not to select some users' needs to implement in the package as the result of a trade-off analysis between their development cost and effect on the market.

NOTE 6 For consumer products, the market segmentation technique can be used to identify sub-groups of users sharing common needs and preference.

The quality needs of the acquirers can be documented as part of their stakeholder needs, along with their owners and necessary evidence.

7.4 Steps for defining quality requirements

7.4.1 Overall description

Quality requirements should be defined clearly and unambiguously, and, where appropriate, quantitatively, in order not to be vague and unverifiable requirements that depend on subjective judgement for their interpretation.

Figure 6 defines an overview of the steps for this purpose for all types of quality requirements.

The steps relate to the following processes and activities in ISO/IEC/IEEE 15288 (See Annex D):

— Requirements definition process

  d) Transform stakeholder needs into stakeholder requirements

  e) Analyze stakeholder requirements

— System requirements definition process

  a) Prepare for system requirements definition

  b) Define system requirements

  c) Analyze system requirements

Based on the quality needs (including feedback from existing systems) obtained from stakeholders and the QIURs, PQRs and DQRs at the higher level of the system hierarchy, QIURs/PQRs/DQRs are defined, analyzed and documented.

Usually these steps are applied iteratively and recursively. When they are applied to a target entity recursively, the steps shall apply to all the ICT products and data within the entity targeted for further engineering activities required for achieving the quality of the target entities.

NOTE 1 From an engineering perspective, defining QIURs precedes defining PQRs/DQRs, which is quite reasonable for developing custom made products, but, in reality, for consumer products, it is often the case that PQRs/DQRs are first defined and then QIURs are defined for evaluating the target entities in operational situations.

For the iteration of the PQR definition steps, to meet some PQRs, technical PQRs should be defined so that they can be used as targets for verification at various stages of development.

During recursive application of the steps, it should also be considered that some functional requirements are derived from some quality requirements (6.5.3).

NOTE 2 In reality, the recursive application of the steps is often carried out concurrently. For instance, the next recursion can be the sub-ICT products and data which constitute the target ICT product (6.5.4). During recursive application of the steps, it should also be considered that some functional requirements are derived from some quality requirements (6.5.3).

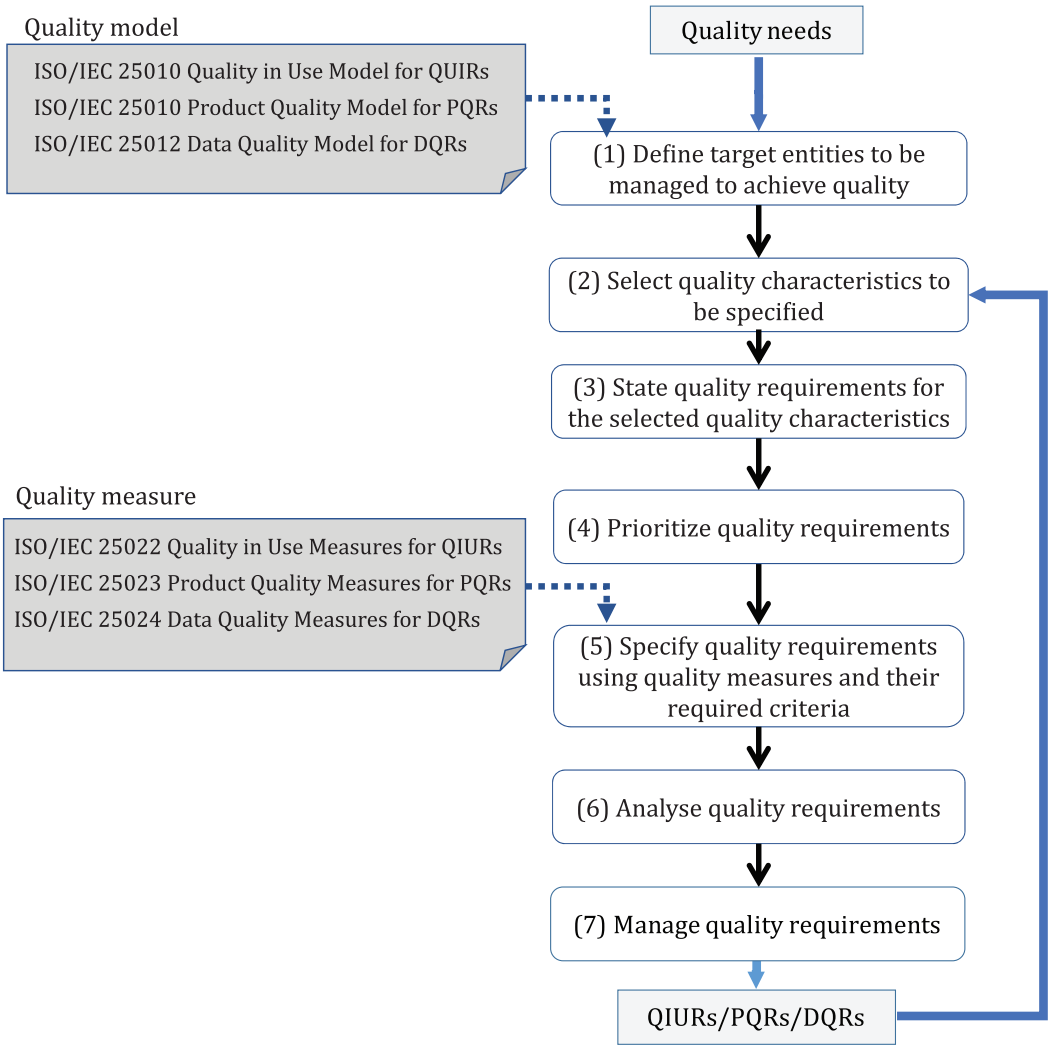

Figure 6 — Steps for defining quality requirements

7.4.2 Definition of steps

Quality requirements for the target entity shall be defined by the following steps.

1) **Define target entities to be managed to achieve quality**

Define the target entity with its boundary, whose quality is to be defined and managed to achieve.

NOTE 1 An ICT product, which can be the target entity of PQRs, can be an arbitrary combination of software, data, hardware and communication. The ICT product includes a system including a plurality of ICT products, a client/server type system, a user PC terminal, a mobile terminal, package software, a database management system and so on.

NOTE 2 Data related to the ICT product can be the target entity of DQRs (6.5.3).

NOTE 3 An information system which can be the target entity of QUIRs includes the target ICT product of PQRs, the users of the ICT product and the relevant environments, which include mechanicals monitored and/or controlled by the ICT product and business processes in which the ICT product is used.

2) **Select quality characteristics to be specified**

For each quality need from stakeholders, determine which quality characteristics (or sub-characteristics) it is classified into. For each QIURs, PQRs and DQRs at the higher level of the system hierarchy, determine which quality characteristics (with how to realize them) required for the target entities to realize them. To do so, use the quality model defined in ISO/IEC 25010 for classification for QIRs and PQRs. Similarly, for DQR, use the quality model defined in ISO/IEC 25012.

NOTE 4 ISO/IEC 25010 and ISO/IEC 25012 can customize the quality model by providing a basis for mapping and customization between the customized model and the standard model. An example of a mapping between a customized model and a standard model can be found in Annex B.

NOTE 5 The quality requirements at the higher level of the system hierarchy includes additional PQRs and DQRs derived from ICT requirements (6.5.1).

NOTE 6 When deriving the quality requirement of the target entity from QIURs, PQRs, and DQRs at the higher level of the system hierarchy, exhaustively examining all the characteristics/sub-characteristics of the quality model could prevent from missing important quality requirements.

NOTE 7 An example of deriving PQRs from QIURs is shown in Annex F.

3) **State quality requirements for the selected quality characteristics**

State quality requirements for the selected quality characteristics so that the following items can be clearly understood:

— Target entity

— Important quality characteristic/subcharacteristic

— User and task (only for QIUR):

— Quality goal with conditions

NOTE 8 Table 3 shows an example of “state PQRs” for a specific system.

Table 3 — Example of "state PQRs"

Target entity	Important quality (sub) characteristic	Quality goal with conditions
Status display function (Display information on delivery cars stored in the database from various kinds of aspects)	Learnability	Operations and displayed contents on this function shall be easy to understand for operators in case that they are normally trained.
	Time efficiency	All displays shall spend under three seconds for being updated after entering any input.
Database component	Availability	The rate of operation shall be large enough to allow 24 hours a day, 365 days a year operation.

4) **Prioritize quality requirements**

Prioritize the derived quality requirements based on their importance and influence for the stakeholders (importance and influence are discussed in 8.1).

NOTE 9 Unlike functional requirements, there are many ways to realize QIURs/PQRs/DRQs at the higher level of the system hierarchy, so quality requirements derived from them can be selectable combinations of multiple candidates, and they are also possible to have some flexible acceptable range. Therefore, it is necessary to prioritize the derived quality requirements from their importance level and influence, so that they can be appropriately selected and defined.

5) **Specify quality requirements using quality measures and their required criteria**

Transform each quality statement into a quality requirement having the following items:

— Quality requirement:

  — Target entity:

  — Selected characteristic:

  — User and task (only for QIUR):

  — Quality goal with conditions:

  — Quality measure:

  — Target value:

  — Acceptable range of values:

To specify the quality measure, use ISO/IEC 25022 for QIUR, ISO/IEC 25023 for PQR and ISO/IEC 25024 for DQR.

NOTE 10 Annex C provides an example for specifying quality requirements.

NOTE 11 ISO 25065 specifies a format and syntax to be used when documenting user requirements, including use-related quality requirements that can include quality in use requirements.

NOTE 12 Because most quality measures with measurement methods are already defined in ISO/IEC 25022, ISO/IEC 25023 and ISO/IEC 25024, quality measures to be used in quality requirements are not necessarily defined from scratch and can be selected from the International Standards.

NOTE 13 Different stakeholders can have different target values depending on their business or engineering needs.

NOTE 14 Regulatory bodies can set minimum/maximum limits for some target values.

6) **Analyze quality requirements**

Analyze the quality requirements to validate them from the following perspectives:

— whether they meet the original needs and requirements of their sources,

— whether they are consistent with the other quality requirements and constraints,

— whether they are verifiable, and

— whether they are feasible,

and resolve problems found.

NOTE 15 See 6.5.5 and Annex G for quality requirements trade-offs.

Conflicts and contradiction among the quality requirements, if found, should be resolved by finding the right balance among them based on their given priorities.

In this step, a risk analysis of each quality requirement should also be performed to identify and resolve risks that the quality requirements could entail. Human, economic, health and safety and environmental risks should be considered selecting specific categories that applies to the problem. And it should be evaluated whether the quality requirements could sufficiently mitigate the risks on the systems at the higher level as expected.

NOTE 16 To execute the risk analysis requirements engineer can work with the users to identify business-related risks that are specific to each quality requirement and, in addition, with the developers to identify the technical risks that are specific to the quality requirement.

7) **Manage quality requirements**

Firstly, obtain explicit agreement on these QIURs and PQRs/DQRs, and they should be approved by all stakeholder groups.

Secondly, establish and maintain traceability between defined quality requirements and their sources (quality needs, QIUR, PQR, and DQR at the higher level).

And finally, if it is decided that the quality requirements need to be improved, perform all the steps iteratively.

8 Using and governing quality requirements

8.1 Critical success factors for implementing quality requirements

The quality requirements of the target entity to be engineered for shall be selected and prioritized based on context of use and the design trade-offs. The quality requirements of the target entity to be verified or validated shall be selected and prioritized based on the critical success factors to meet the stakeholders' objectives.

NOTE 1 Quality requirements serve two purposes: a) To guide and prioritise design solutions that are expected to meet the quality requirements, and b) To provide acceptance criteria that can be evaluated

Since minimizing development cost and time is also important in reality, the processes for engineering, and verification and validation should be both effective and efficient, and in some cases, some compromise will be needed. Therefore a risk-based approach is recommended, which has the following steps.

1) Evaluate each quality requirement and give priorities to it on the following points.

   — Importance: obligation for success-critical stakeholders, and criticality to society, business, human lives and/or environment.

   — Influence: impact on the development and maintenance process by causing rework.

2) Plan activities and the points of their enforcement (development phases) for verification and validation for the quality requirement, and estimate their cost and effect. Such activities include testing, inspection, prototyping, adoption of effective design methods, iterative processes, etc.

3) On each quality requirement with high priority, trade-off the evaluated risks of not-achieving it vs. the cost for performing the activities to prevent the risk from occurring. If the activities are considered cost-effective, then adopt the activities and incorporate them into the development plan.

NOTE 2 When the product contains a Data Base Management System (DBMS) software component, the DQRs for the data handled by it are greatly simplified and could be absent.

8.2 Quality requirements traceability

Bi-directional traceability between quality requirements and ICT components shall be maintained and reaffirmed throughout the lifecycle of the product with:

— elements of product implementation, design and evaluation artifacts, and

— stakeholder requirements and system requirements.

Example of implementing PQRs/DQRs traceability in the product development lifecycle:

1) Functional requirements; such as security requirement and access control function which realizes it.

2) Architecture; such as fault tolerant requirement and architecture which realizes it.

3) PQRs/DQRs for deployed components; such as response time requirement for software and response time requirements for components of the software.

4) Policies for design process; such as security requirement and secure coding policy which realizes it.

NOTE Annex H shows an example of quality requirements traceability in development stages.

8.3 Critical factors for testing quality requirements

Testing should be conducted based on the defined quality requirements in order to quantify the quality characteristics and subcharacteristics of the target system. For example, to quantify the functional suitability of the target system, functional testing is conducted, and to quantify performance efficiency, performance test is performed. To quantify consistency of data, comparison of data is conducted and to quantify completeness of data a count of null values is performed.

NOTE 1 ISO/IEC 25040 provides the requirements and recommendations for evaluation process using quality requirements.

Testing can be conducted on various stages of development and are reflective of the different quality requirements. For example, software and system testing can be performed based on PQRs/DQRs and acceptance testing on QIURs, and unit and integration testing on technical PQRs/DQRs.

NOTE 2 Technical PQRs are requirements for technically identified attributes (targeting specifications, source code, etc.) to meet the other PQRs, which can be used to verify the intermediate products by inspection and testing at various stages of development and maintenance.

Testing normally involves sequences of processes starting with planning the test, designing and developing test cases related to specific quality measures, followed by executing the test cases. Report (incident/defect report and test report) will then be produced based on the result of the testing and other findings depending on test objectives (as specified in planning document).

NOTE 3 The process of testing or software testing is specified and explained in ISO/IEC/IEEE 29119-2.

Annex A
(informative)

Recommended process for elicitation of quality needs

A.1 General

This annex describes the process proposed to elicitation of quality needs. The phases of the process and the steps that the quality engineer should execute in order to properly extract and define quality needs are discussed in detail. It is important that the quality engineer should consider all the needs of the various stakeholders of the product.

This process is designed to be complementally integrated in ISO/IEC/IEEE 15288.

Initially, the different steps of the process and the order in which they should be executed are presented in a general perspective. For each step, the necessary input the quality engineer should use and the output the step should produce are specified. Thereafter, the characteristics of each step of this process will be specified in more details.

There are two main phases of this process, namely:

1) Defining the project context.

2) Defining quality needs of each stakeholder.

A.2 Defining the project contexts [S1]

A.2.1 General

In this phase of the process, the contexts of the project are defined considering several factors like the necessary assumptions, the main characteristics of the project domain, the constraints of the project and the list of the stakeholders. See Figure A.1. This list of stakeholders is crucial for the execution of rest of the process because it will be used in the second phase to specify the ICT product quality requirements. Possibly the list of stakeholders will be incomplete at this stage. New stakeholders will appear during the analysis process.

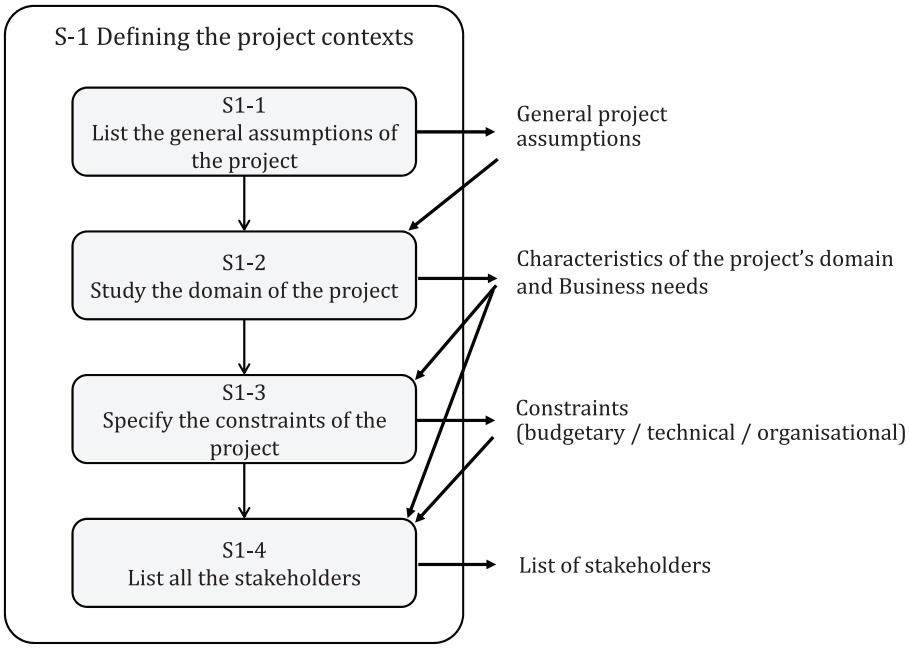

Figure A.1 — Defining the project contexts

A.2.2 List the general assumptions of the project [S1-1]

Input: None

Output: General project assumptions

The first step of this phase is to consider the following assumptions in the specific context of the project:

— The customer/user is a specialist in his area of business.

— The customer/user is unlikely to be familiar with the concepts of system/software quality.

— The quality engineer is specialized in the discovering and specification of quality requirements and unlike to be an expert in the area of business of the customer/user.

— Any other necessary assumption from the perspective of quality that is relevant to the project.

Therefore, the quality engineer should consider all these factors before starting the process and should adapt to them in order to adequately specify the right quality requirements. As specified below, the process of specifying quality requirements is based on the determination of the quality needs that support the business needs of the different identified stakeholders. The more exhaustively the quality engineer identifies the necessary assumptions of the project the better can he collaborate with the customer/user and define the quality needs of the different stakeholders. Thus, the quality engineer responsible for specifying quality needs from stakeholders should keep in mind this list of assumptions to complete the system/software quality requirements specification.

A.2.3 Study the domain of the project [S1-2]

Input: General project assumptions

Output: Characteristics of the project's domain, business needs

This step is essentially used to have a better understanding of the project domain and the general context in which the ICT product will have to be deployed. This step is very important in the process of specifying quality requirements since it allows the quality engineer to gain more expertise in the project domain and to better understand the customer's/user's business needs. In addition, this step is

essential to determine the feasibility of the various aspects of the project because it gives a clear picture of the resources (financial capital and technology infrastructure) and skills (staff and knowledge) necessary to achieve the ICT product quality required by the customer/user.

Consequently, the execution of this step will allow the quality engineer to better understand the different aspects of the area in which the project will occur and to facilitate the communication with the different stakeholders, which is essential to specify the system/software quality requirements.

A.2.4 Specify the constraints of the project [S1-3]

Input: Characteristics of the project's domain, business needs

Output: Budgetary, technical, organizational and other constraints

During this step, the quality engineer should use his understanding of the project domain resulting from the previous step in order to specify the constraints of the project. This step should involve both of the customer/user and the different stakeholders within the ICT product supplier team.

This step is used to define three main categories of constraints:

— Budgetary constraints, which will depend essentially on the financial resources allocated for the purpose of the quality of the ICT product.

— Technical constraints, which are initially stated by the development team from the software supplier. Moreover, in the case where the customer/user has to perform the system/software maintenance after its deployment, it is important to involve his technical team in this step. In addition, the infrastructure, which the customer/user has at his disposal will necessarily impose some technical limits to the development team.

— Organizational constraints, which are defined primarily by the structure of customer's/user's company and the various interactions that exist within it. Therefore, the activity of specifying ICT product quality requirements should take into account these constraints, so they can be well integrated into the project execution and further into the business model supported by the ICT product.

Finally, the quality engineer should identify other types of constraints that are relevant to a specific project with the help of the customer/user or the development team. The complete list of constraints is decisive to evaluate the feasibility of system/software quality requirements specified by the engineer.

A.2.5 List all the stakeholders [S1-4]

Input: The characteristics of the project domain, constraints

Output: List of stakeholders

The final step of the first phase of the process has the objective to establish the list of the stakeholders of the project. Once the quality engineer has a better understanding of the project and its constraints, it is very important to identify all stakeholders from both sides: the customer/user and the supplier. Missing some stakeholders could negatively impact the quality of the final ICT product, which, in consequence, could lead to a failure of the project.

There needs to be an initial high-level activity to agree who the stakeholders are, the relative priorities for how the product will/could support different stakeholders, and other aspects of the context of use

The quality engineer should closely collaborate with the customer/user to specify his technical and nontechnical stakeholders.

NOTE As defined in ISO/IEC 25010, the stakeholders of the ICT product include users and other stakeholders (like developer, regulatory body, and society), and users can be divided into 3 categories: primary, secondary and indirect.

In addition, the quality engineer should consider the ICT product supplier stakeholders. This list of stakeholders will be essential for the specification of ICT product quality requirements in the next phase of the process. Once the list of the stakeholders is established and validated with the customer/user, the quality engineer should be able to proceed to the next phase of the process.

A.3 Defining quality needs of each stakeholder [S2]

A.3.1 General

The purpose of this phase is to specify the quality requirements of each stakeholder identified in the previous phase. The first steps in this phase extract the quality needs of each of identified stakeholders. Subsequently, these needs are used in the process of specifying quality requirements and the measures. Finally, all the requirements of each stakeholder are analyzed to resolve conflicts, to perform a risk analysis and to validate them with the stakeholder. See Figure A.2.

Therefore, the steps of this phase are necessary to be performed in a loop for each stakeholder determined in the previous phase of the process (activities described in A.3.2 to A.3.4).

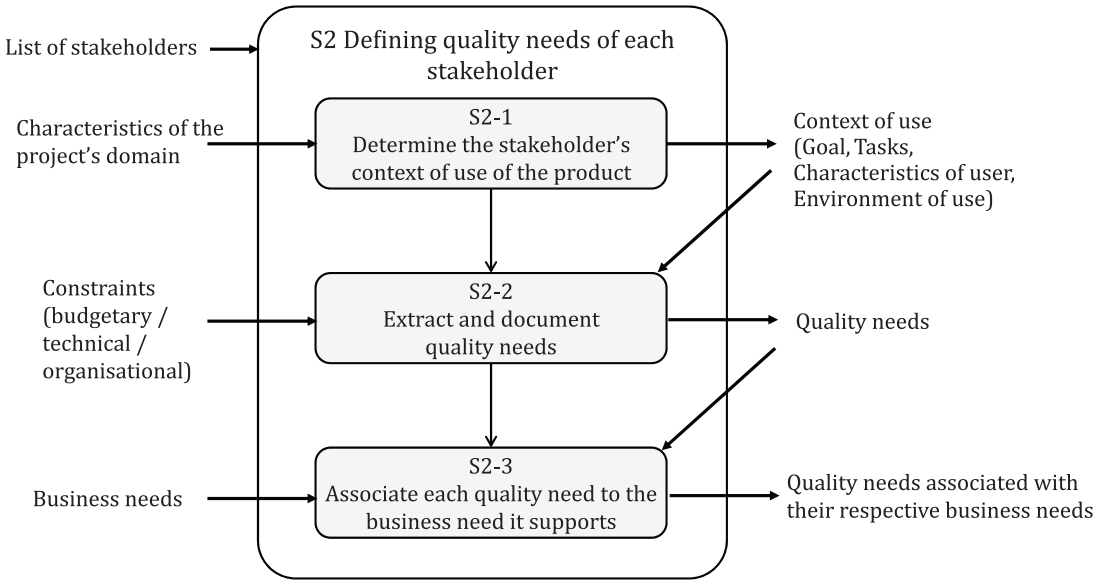

Figure A.2 — Defining quality needs of each stakeholder

A.3.2 Determine the stakeholder's context of use of the product [S2-1]

Input: List of stakeholders, Characteristics of the project domain

Output: Contexts of use related to identified stakeholders

This step should be executed in collaboration with the stakeholder. Its purpose is to determine the context of use of the ICT product specific for a given stakeholder or user group. To achieve this, the quality engineer should determine:

— The goal that the user wants to achieve through the use of the ICT product.

— The tasks that the user will perform in order to achieve his goal.

— The characteristics of the user,

— The technical, physical and organizational environment of use.

The understanding of the stakeholder's context of use can help the quality engineer to determine resulting quality needs later in the process. Considering that any stakeholder usually understands well his own context of use, his collaboration in this step is crucial to assure its success.

In order to define the context of use, the quality engineer can use a large number of techniques that allow him to obtain this information:

— survey;

— observation;

— interview;

— any other tool that fits the situation.

In some cases it is preferable to combine several of these techniques in order to obtain more information and have a stronger basis for specifying system/software quality requirements thereafter.

Finally, the quality engineer can refer to ISO/IEC 25063 to document the context of use of each stakeholder to ensure its traceability and to adopt a common industrial format.

NOTE 1 ISO/IEC 25063 can be used not only for documenting, but also for identifying the type of information to be collected

Tasks that the user will perform in order to achieve his goal should be examined carefully because these tasks usually contain most of quality attributes, like, for example, localized performance or usability for a specific task.

NOTE 2 High-level goal analysis does not show this specific quality needs.

A.3.3 Extract and document quality needs [S2-2]

Input: Context of use, constrains

Output: Quality needs

The quality engineer should use the information he obtained from the context of use provided by each stakeholder or user in the stakeholder's organization in order to identify his quality needs. The stakeholder's context of use should give a clear idea of his quality needs and allow for revising their scope. Therefore, in collaboration with the stakeholder, the engineer should list all resulting quality needs, verify and validate them to ensure that they represent his real needs.

Further, in order to correctly identify and define all relevant quality needs, the quality engineer should consider all the tasks necessary to achieve stakeholder's goal but also the environment in which these tasks are executed.

The quality engineer is required to document all the quality needs in a format that enables them to be traced back to their sources.

A.3.4 Associate each quality need to the business need it supports [S2-3]

Input: Quality needs, business needs

Output: Quality needs associated with their respective business needs

This step is not necessarily decisive for the conduct of the rest of the process but it is fundamental in order to justify the effort required to implement the identified quality needs, therefore the quality engineer is required to explicitly specify for each identified quality need the business need it originates from.

Once this step is completed, the quality engineer should be able to initiate the next step which is central in this process. This step allows him to specify the ICT product quality requirements for each stakeholder using the quality needs that he has defined earlier.

NOTE The context mentioned in A.2.3 includes understanding of the business model in which the ICT product is included, or at least the portion of the business model in which the ICT product under analysis will operate.

Annex B
(informative)

Example for mapping quality needs to quality characteristics

B.1 General

This annex provides an example for mapping quality needs to quality characteristics of ISO/IEC 25010.

Some quality needs cannot be mapped directly into specific characteristics on the model, but are required to state and specify. The following process describes how to define quality requirements based on such quality needs, combining several characteristics/sub-characteristics of the ISO/IEC 25010 model.

For each quality need that cannot be stated and specified directly mapping the need to ISO/IEC 25010 unique quality characteristics, existing characteristics/sub-characteristics could be used as building blocks for the new one.

NOTE 1 New characteristics/measures could differ from the simple union/addition of their components. They express new quality characteristics to be included in the product.

NOTE 2 Maintaining the discipline of correctly using characteristics/sub-characteristics and measures, models could be extended and tailored to diverse and multiple situations. The basic characteristics proposed by ISO/IEC 25010 can be used as building blocks to represent more complex quality characteristics if required by the products.

B.2 Example: Mapping the “Hand over control” quality need

First quality needs are elicited by using Annex A.

Product

A self-drive car, fully autonomous.

Project

A product manager states customer quality needs related to a self-drive vehicle. Stated below is one of these functional needs:

The vehicle makes the decision to hand over control to the human driver based on its own situation recognition, which could be risky in a complex situation, for an inexperienced driver or even for an experienced one.

Product manager states quality needs on “Hand over control” as follows:

QN1: Minimize the number of handing over control from the vehicle to human driver for the scenarios under considerable difficult situations.

The ISO/IEC 25010 and ISO/IEC 25012 quality models can be used to state quality characteristics. Because using the models can help define unambiguous and verifiable quality requirements, the organizations to do so will have a better position to win competition with quantifiable and demonstrable quality of the product.

Context of the new product

- Immature/new technology
- Immature/new regulations
- Lack of buyer's knowledge about the product

Domain

- Highly regulated market, but immature in regards to this new technology
- Standards under development
- Regulations vary geographically/regulations under development
- Diversity of customers
- Diversity of needs
- Diversity of customer's flavours
- Different geographical zones, society and cultures
- Many suppliers
- Many contributors to the business model: insurance, unions, etc.

Stakeholders

- Regulatory bodies
- Producers
- Clients/Buyers
- Auto part industry
- All kind of car repairing services
- Commercial agencies
- Insurance companies
- Transportations (people and merchandise)
- Unions
- …

Next, the quality needs defined above are mapped into some of quality characteristics/ subcharacteristics. But the high-level quality need like QN1 cannot be directly mapped to any single characteristic/ subcharacteristic in the ISO/IEC 25010 quality models (and its measure defined in ISO/IEC 25022, ISO/IEC 25023 and ISO/IEC 25024). For this situation, a set of subcharacteristics can be used to specify quality requirements for the need.

To specify quality requirements for QN1, a combination of the subcharacteristics in the quality in use model and their measures can be used. See Table B.1.

Table B.1 — Example of mapping QN1 into subcharacteristics in quality in use model

Subcharacteristic	Measure
Effectiveness	Objectives achieved
Trust	User trust
Health and safety risk	Safety of people affected by use of the system
Context completeness	Context completeness

Table B.1 *(continued)*

Subcharacteristic	Measure
Flexibility	Flexible context of use

NOTE Quality requirements on some characteristic/sub-characteristic can have different measures and target values, depending on different stakeholder's concerns. Such a situation can be resolved with technical and business decisions.

Annex C
(informative)

Example for specifying quality requirements

Figure C.1 depicts a framework for specifying quality requirements with an example of the maturity requirement of.

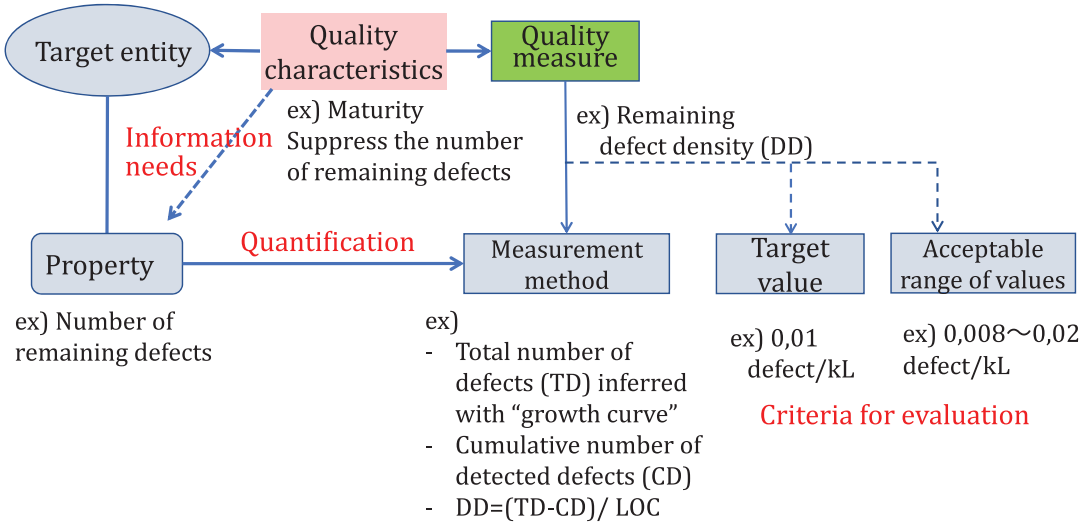

Figure C.1 — Example for specifying quality requirements

The requirement of the example shown in Figure C.1 can be specified as follows:

— Quality requirement:
  — Target entity: XX software
  — Selected characteristic: Maturity
  — Quality goal with conditions: The estimated number of remaining defects shall be suppressed
  — Quality measure: Remaining defect density
  — Target value: 0,01 defects/kL
  — Acceptable range of values: 0,008- 0,02 defects/kL

Annex D
(informative)

Relationship to ISO/IEC/IEEE 15288 (System lifecycle processes)

Table D.1 shows the relationship between the steps for defining quality requirements in this document and the requirements-related processes defined in ISO/IEC/IEEE 15288.

Table D.1 — Relationship between the steps for quality requirements in this document and the requirements-related processes of ISO/IEC/IEEE 15288:2015

ISO/IEC/IEEE 15288: 2015			ISO/IEC 25030
6.4.2 Stakeholder needs and requirements definition process			**Clause 7 Quality requirements processes**
Activity	a)	**Prepare for Stakeholder Needs and Requirements Definition.**	
Task	1)	Identify the stakeholders who have an interest in the system throughout its life cycle.	7.3.1 Identification of stakeholders
Task	2)	Define the stakeholder needs and requirements definition strategy.	
Task	3)	Identify and plan for the necessary enabling systems or services needed to support stakeholder needs and requirements definition.	
Task	4)	Obtain or acquire access to the enabling systems or services to be used.	
Activity	b)	**Define stakeholder needs.**	
Task	1)	Define context of use within the concept of operations and the preliminary life cycle concepts.	7.3.2 Defining stakeholder needs
Task	2)	Identify stakeholder needs.	
Task	3)	Prioritize and down-select needs.	
Task	4)	Define the stakeholder needs and rationale.	
Activity	c)	**Develop the operational concept and other life cycle concepts.**	
Task	1)	Define a representative set of scenarios to identify all required capabilities that correspond to anticipated operational and other life cycle concepts.	
Task	2)	Identify the interaction between users and the system.	
Activity	d)	**Transform stakeholder needs into stakeholder requirements.**	
Task	1)	Identify the constraints on a system solution.	
Task	2)	Identify the stakeholder requirements and functions that relate to critical quality characteristics, such as assurance, safety, security, environment, or health.	7.4.2 (2) Select quality characteristics to be specified (Quality in use)

Table D.1 *(continued)*

ISO/IEC/IEEE 15288: 2015			ISO/IEC 25030
Task	3)	Define stakeholder requirements, consistent with life cycle concepts, scenarios, interactions, constraints, and critical quality characteristics.	7.4.2 (3) State quality characteristics (Quality in use)
Activity	e)	**Analyze stakeholder requirements.**	
Task			7.4.2 (4) Prioritize quality requirements (QIURs)
Task	1)	Analyze the complete set of stakeholder requirements.	7.4.2 (5) Specify quality requirements using measures and their required criteria (QIURs) 7.4.2 (6) Analyze quality requirements (QIURs)
Task	2)	**Define critical performance measures** that enable the assessment of technical achievement.	
Task	3)	Feedback the analyzed requirements to applicable stakeholders to validate that their needs and expectations have been adequately captured and expressed.	
Task	4)	Resolve stakeholder requirements issues.	
Activity	f)	**Manage the stakeholder needs and requirements definition.**	
Task	1)	Obtain explicit agreement on the stakeholder requirements.	7.4.2 (7) Manage quality requirements (QIURs)
Task	2)	Maintain traceability of stakeholder needs and requirements.	
Task	3)	Provide key information items that have been selected for baselines.	
ISO/IEC/IEEE 15288: 2015 **6.4.3 System requirements definition process**			
Activity	a)	**Prepare for System Requirements Definition.**	
Task	1)	Define the functional boundary of the system in terms of the behavior and properties to be provided.	7.4.2 (1) Define target entities to managed to achieve quality
Task	2)	Define the system requirements definition strategy.	
Task	3)	Identify and plan for the necessary enabling systems or services needed to support system requirements definition.	
Task	4)	Obtain or acquire access to the enabling systems or services to be used.	
Activity	b)	**Define system requirements.**	
Task	1)	Define each function that the system is required to perform.	
Task	2)	Define necessary implementation constraints.	
Task	3)	Identify system requirements that relate to risks, criticality of the system, or critical quality characteristics.	7.4.2 (2) Select quality characteristics to be specified (Product/Data quality)
Task	4)	Define system requirements and rationale.	7.4.2 (3) State quality characteristics (Product/Data quality)
Activity	c)	**Analyze system requirements.**	
Task			7.4.2 (4) Prioritize quality requirements (PQRs/DQRs)

Table D.1 *(continued)*

ISO/IEC/IEEE 15288: 2015			ISO/IEC 25030
Task	1)	Analyze the complete set of system requirements.	7.4.2 (5) Prioritize quality requirements (PQRs/DQRs)
Task	2)	**Define critical performance measures** that enable the assessment of technical achievement.	
Task	3)	Feedback the analyzed requirements to applicable stakeholders for review.	
Task	4)	Resolve system requirements issues.	
Activity	d)	**Manage system requirements.**	
Task	1)	Obtain explicit agreement on the system requirements.	7.4.2 (5) Specify quality requirements using measures and their required criteria (PQRs/DQRs) 7.4.2 (6) Analyze quality requirements (PQRs/DQRs)
Task	2)	Maintain traceability of the system requirements.	
Task	3)	Provide key information items that have been selected for baselines.	

Annex E
(informative)

Relationship to ISO/IEC/IEEE 29148 (Requirement engineering)

E.1 Processes and documentations of quality requirements

ISO/IEC/IEEE 29148 defines requirements engineering processes, strictly complying with the two technical processes of ISO/IEC/IEEE 15288. See Figure E.1.

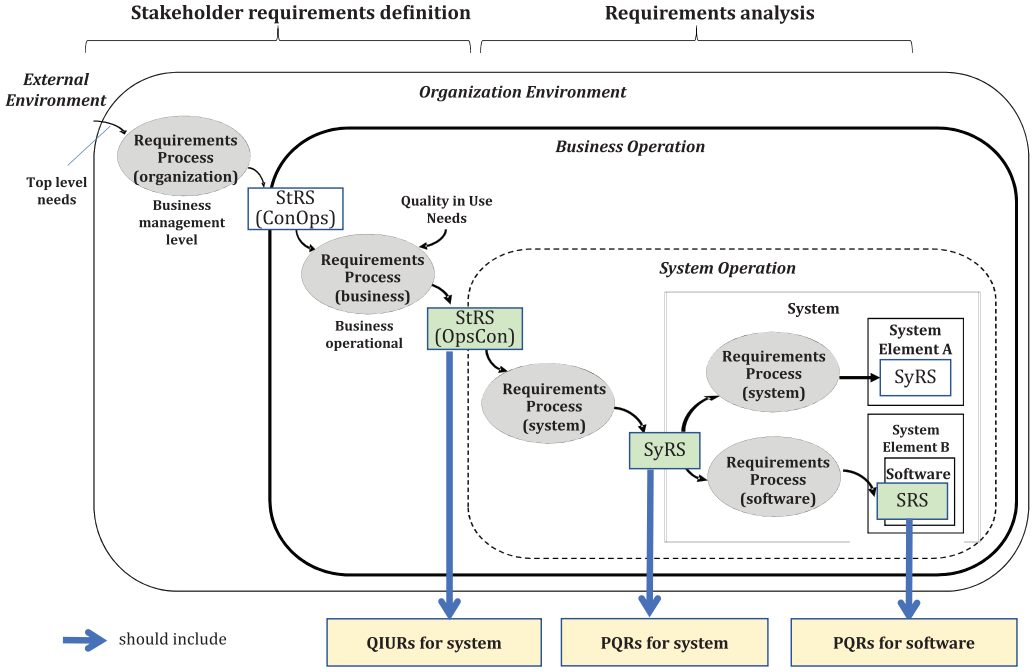

Figure E.1 — Relationship to ISO/IEC/IEEE 29148

The requirements engineering processes produce:

— StRS: Stakeholder requirements specification

— SyRS: System requirements specification

— SRS: Software requirements specification

They are respectively corresponding to the following quality requirements:

— Quality in use requirements for system

— Product quality requirements for system

— Product quality requirements for software

Table E.1 shows where these quality requirements are positioned in the requirements types in ISO/IEC/IEEE 29148.

Table E.1 — Important examples of the requirements type attribute in ISO/IEC/IEEE 29148

Requirements type	Description	
Functional	Functional requirements describe the system or system element functions or tasks to be performed.	
Performance	A requirement that defines the extent or how well, and under what conditions, a function or task is to be performed. These are quantitative requirements of system performance and are verifiable individually. Note that there may be more than one performance requirement associated with a single function, functional requirement, or task.	
	Usability/Quality-in-Use Requirements (for user performance and satisfaction)	Provide the basis for the design and evaluation of systems to meet the user needs. Usability/Quality-in-Use requirements are developed in conjunction with, and form part of, the overall requirements specification of a system.
Interface	Interface requirements are the definition of how the system is required to interact with external systems (external interface), or how system elements within the system, including human elements, interact with each other (internal interface).	
Design Constraints	A requirement that limits the options open to a designer of a solution by imposing immovable boundaries and limits (e.g., the system shall incorporate a legacy or provided system element, or certain data shall be maintained in an on-line repository).	
Process requirements	These are stakeholder, usually acquirer or user, requirements imposed through the contract or statement of work. Process requirements include: compliance with national, state or local laws, including environmental laws; administrative requirements; acquirer/supplier relationship requirements; and specific work directives. Process requirements may also be imposed on a program by corporate policy or practice. System or system element implementation process requirements, such as mandating a particular design method, are usually captured in project agreement documentation such as contracts, statements of work, and quality plans.	
Non-Functional	These specify requirements under which the system is required to operate or exist or system properties. They define how a system is supposed to *be.* Quality requirements and human factors requirements are examples of this type.	
	Quality Requirements *NOTE 1 Additional guidance on quality requirements can be found in the ISO/IEC SQuaRE standards, especially ISO/IEC 25030 and in ISO/IEC 25010.*	Include a number of the 'ilities' in requirements to include, for example, transportability, survivability, flexibility, portability, reusability, reliability, maintainability, and security. The list of non-functional, quality requirements (e.g., “ilities”) should be developed prior to initiating the requirements document. This should be tailored to the system(s) being developed. As appropriate, measures for the quality requirements should be included as well.
	Human Factors Requirements	State required characteristics for the outcomes of interaction with human users (and other stakeholders affected by use) in terms of safety, performance, effectiveness, efficiency, reliability, maintainability, health, well-being and satisfaction. These include characteristics such as measures of usability, including effectiveness, efficiency and satisfaction; human reliability; freedom from adverse health effects.

E.2 Mapping quality requirements to the recommended outline

This annex gives an example of how ISO/IEC 25010 quality characteristics could map to the ISO/IEC/IEEE 29148 defined document outline [Stakeholder requirements specification (StRS), System requirements specification (SyRS) and Software requirements specification (SRS)].

The recommended document outline of StRS mostly addresses the Quality in Use of ISO/IEC 25010 as shown in Table E.2.

Table E.2 — Mapping of Quality In Use to StRS, SyRS and SRS

Quality In Use	ISO/IEC/IEEE 29148 documents		
	StRS	SyRS	SRS
Effectiveness	9.3.15 User requirements 9.3.6 Goal and Objectives	9.4.5 Usability requirements	—
Efficiency	9.3.9 Business processes 9.3.10 Business operational policies and rules 9.3.11 Business operational constraints 9.3.12 Business operation modes 9.3.13 Business operational quality 9.3.15 User requirements 9.3.16 Operational concept 9.3.17 Operational scenarios	9.4.5 Usability requirements 9.4.6 Performance requirements	—
Satisfaction	9.3.4 Stakeholders 9.3.5 Business environment 9.3.9 Business processes 9.3.10 Business operational policies and rules 9.3.11 Business operational constraints 9.3.12 Business operation modes 9.3.13 Business operational quality 9.3.14 Business structure 9.3.15 User requirements	9.4.5 Usability requirements 9.4.8.1 Human system integration requirements	—
Freedom from risk	9.3.5 Business environment 9.3.7 Business model 9.3.8 Information environment 9.3.18 Project constraints	9.4.14 Policies and regulations	—
Context Coverage	9.3.1 Business purpose 9.3.2 Business scope 9.3.6 Goal and Objective	9.4.1 System purpose 9.4.2 System scope 9.4.3.1 System context	—

The recommended document outline of SyRS and SRS mostly address the Product Quality of ISO/IEC 25010 as shown in Table E.3.

Table E.3 — Mapping of Product Quality to StRS, SyRS and SRS

Product Quality	ISO/IEC/IEEE 29148 documents		
	StRS	SyRS	SRS
Functional Suitability	—	9.4.4 Functional requirements	9.5.6 Limitations i) Quality Requirements 9.5.11 Functions

Table E.3 *(continued)*

Product Quality	ISO/IEC/IEEE 29148 documents		
	StRS	**SyRS**	**SRS**
Performance Efficiency	—	9.4.6 Performance requirements	9.5.3.6 Memory constraints 9.5.6 Limitations i) Quality Requirements 9.5.13 Performance requirements
Compatibility	—	9.4.7 System interfaces	9.5.3.1 System interfaces 9.5.3.5 Communications interfaces 9.5.6 Limitations i) Quality Requirements
Usability	—	9.4.5 Usability requirements 9.4.8.1 Human system integration requirements	9.5.3.2 User interfaces 9.5.3.7 Operations 9.5.6 Limitations i) Quality Requirements 9.5.12 Usability requirements
Reliability	—	9.4.8.3 Reliability	9.5.6 Limitations i) Quality Requirements 9.5.17 a) Reliability 9.5.17 b) Availability
Security	—	9.4.12 System security	9.5.6 Limitations i) Quality Requirements 9.5.17 c) Security
Maintainability	—	9.4.8.2 Maintainability	9.5.6 Limitations i) Quality Requirements 9.5.17 d) Maintainability
Portability	—	9.4.9 System modes and states 9.4.10.1 Physical requirements 9.4.10.2 Adaptability 9.4.11 Environmental conditions	9.5.3.3 Hardware interfaces 9.5.3.4 Software interfaces 9.5.6 Limitations i) Quality Requirements 9.5.17 e) Portability

Annex F
(informative)

Derivation from quality in use requirements to product quality requirements

A QIUR can imply several PQRs as follows:

— Effectiveness, efficiency and satisfaction requirements by considering the business operations using the target product. For example, an efficiency requirement can imply time efficiency, usability, functional correctness and interoperability requirements for the product since these product qualities affects together on efficiency of the business operations.

— Freedom to risk requirements by considering the scenarios on wrong or malicious uses of the target product and trouble occurrences of itself. A freedom to risk requirement can imply functional correctness, reliability, security, usability and maintainability requirements.

— Context coverage requirements by considering the scenarios on a variety of use contexts including different kinds of users and changes of the operational environment. A context coverage requirement can imply usability, compatibility, maintainability and time efficiency requirements.

There are a lot of derivation patterns from quality in use characteristics to product quality characteristics, depending on system category. Table F.1 shows an example of the derivation patterns for “nuclear reactor control system”.

Table F.1 — Example derivation from QIURs to PQRs for “nuclear reactor control system”

QIUR	Derivated PQR	
	characteristic/ subcharacteristic	**Statement**
Freedom from risks Even if the interior of the nuclear reactor is in a "dangerous state", it will not result in an accident of level 6 or higher. (Probability of accident 10^{-9})	**Maturity**	The number of defects of the nuclear reactor control system shall be minimized
	Fault tolerant	The nuclear reactor control system shall continue operation without going into an uncontrollable state against any combination of failures
	Capacity	The margin of performance of the nuclear reactor control system shall be ensured.
	Functional correctness	The situation inside the nuclear reactor shall be accurately displayed.
	Time efficiency	The situation inside the nuclear reactor shall be quickly displayed.
	Operability	The nuclear reactor control functions shall be easy to operate.
	User error protection	When the inside of the nuclear reactor is in the "dangerous state", the nuclear reactor control system shall support the operators to execute the operation of quickly returning to the normal state without mistake.

The nuclear reactor control system should have a strong freedom from risk requirement. Since the system is categorised into a realtime and human-intensive system, not only reliability but also

performance efficiency and usability are important. Considering category of the target system using ISO/IEC TR 12182 can lead to a better derivation.

NOTE To some extent the relationships depend on the task, e.g. maintainability and portability will affect effectiveness, efficiency and satisfaction for maintenance and porting tasks.

Annex G
(informative)

Example of relationship between product quality characteristics

This annex provides an example of relationship between any two types of product quality characteristics.

Table G.1 shows an example of the relationship that might exist between product quality characteristics.

Table G.1 — Example of relationship between product quality characteristics

	Functional suitability	Reliability	Performance efficiency	Usability	Security	Compatibility	Maintainability	Portability
Functional suitability		–	–	–	–	–	–	–
Reliability					+		+	
Performance efficiency				+			–	
Usability			–		–			
Security	–	–	–	–		–	–	–
Compatibility	+		–		–			
Maintainability			–		–			+
Portability			–		–	+		

Key

\+ : positive effects (the quality characteristic in the row can positively influence on that in the column)

–: negative effects (the quality characteristic in the row can negatively influence on that in the column, meaning conflicts can occur)

The following explanations are given for each row of the table.

— To add some functions for better functional completeness (functional suitability) requirements can cause some problems on the quality for all the other quality characteristics.

— To increase reliability (maturity) of some component can help achieve security and maintainability for the overall system.

— To realize some performance efficiency (time efficiency) requirements for a set of functions can result in better usability (operability for the functions), while it can lead to a worse maintainability (reusability, modularity of the components).

— The infinite undo capability to achieve a better user error protection can need a large amount of memory for its undo buffer (performance efficiency-resource utilization), and can increase possibility to steal some information from the buffer (security-confidentiality).

— To achieve better security can cause negative effects on the other characteristics, for example, to introduce many authentifications can badly affect usability (operability).

— To increase interoperability with the other systems can contribute to the better functional suitability by collaborating with them, while it can cause some performance efficiency problems and can introduce some security flaws due to its heavy and vulnerable communication protocol.

— To achieve a better testability by using a simple stream interface can need additional computation time for text processing (performance efficiency-time efficiency), can allow easy manipulation

of the information through the interface (security-authentification), while it can contribute to simplification of initial testing which enables smoother installation.

— To achieve a better adaptability for many platforms can cause some performance and security problems on some of the platforms, while it can contribute to a better compatibility (interoperability).

NOTE The relationships are not always symmetric.

To find conflicts between two requirements, the two cells which show the relationships between the quality characteristics of the requirements should be checked.

Annex H
(informative)

Example of deployment and traceability of quality requirements to software

PQRs/DQRs are deployed into software components (including data components) as shown in Figure H.1. There are four types of deployment:

a) Functional requirements to implement quality requirements, e.g., security requirement -> access control functional requirements.

b) Architecture to implement quality requirements, e.g., fault tolerant requirement -> fault tolerant architecture.

c) Quality requirements for deployed components, e.g., response time requirement -> response time requirements for its components.

d) Policies for design process, e.g., security requirement -> secure coding policy.

Figure H.1 also describes the necessity of the bi-directional traceability around the PQRs/DQRs for the target software to be established and maintained.

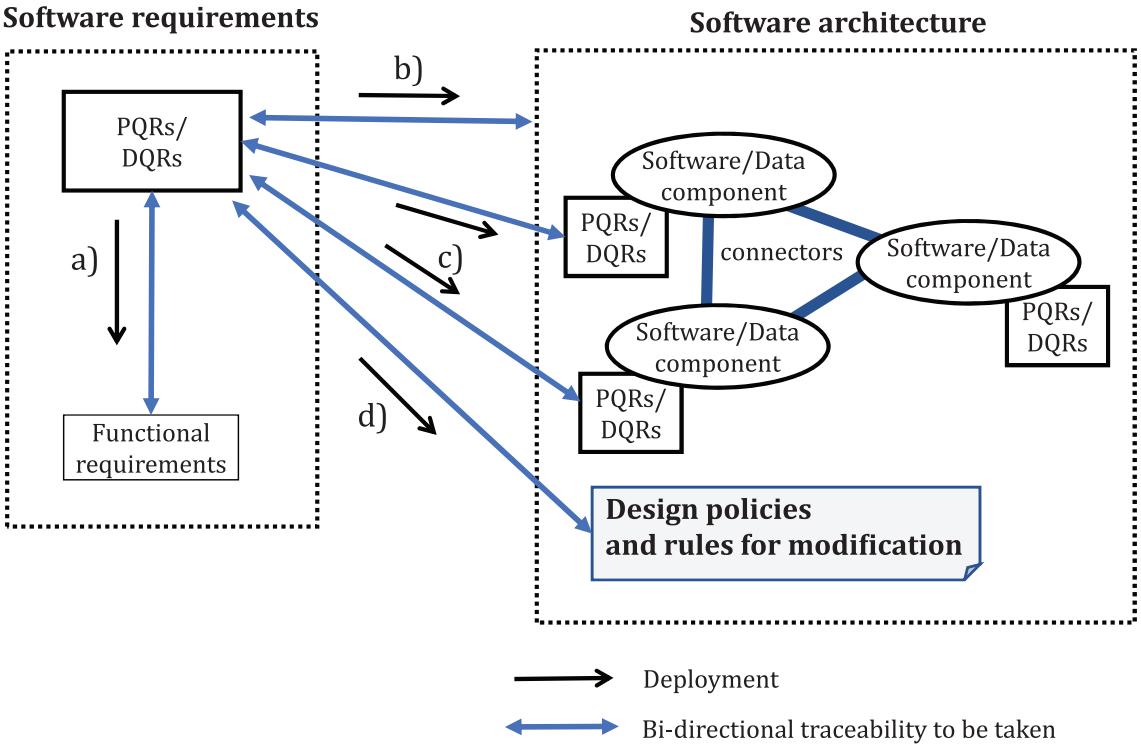

Figure H.1 — Example of quality requirements traceability in development stages

Annex I
(informative)

Example of stakeholder-target matrix

Table I.1 shows some of the important quality requirements for certain users and tasks on the case of internet shopping.

Table I.1 — An example of stakeholder-target matrix for users on the case of internet shopping

Target entity	User				
	Primary user		Secondary user		Indirect user
	Internet shopper[*1]	*Operator*[*2]	*Helper*[*1]	*Manager of buyer department*[*2]	*Owner*[*2]
Information system	When he finishes using the system; **Satisfaction** When he enter his credit card number; **Freedom from risks**	When he answers to questions from shoppers; **Effectiveness, Efficiency** When he changes/ delete shoppers 's order; **Freedom from risks**	When he answers to questions from shoppers; **Effectiveness, Efficiency** When he changes/delete user's order; **Freedom from risks**	On the business process of his department; **Effectiveness, Efficiency**	For all assets, scenes and threats; **Freedom from risks**
ICT product	When he searches and buys items using a browser; **Functional correctness, Usability**	When he uses XX terminal; **Usability**			

*1 User-1: Internet shopping customers, who browse shops, selecte goods, and act for buying something.

— **Primary use**r: Internet shopper, who searches for goods, selects and decides and orders goods by using a computer through internet.

— **Secondary user**: Helper, who helps primary users to use the system.

— **Indirect user**: Customer, who wants something to buy, asking someone to do internet shopping, not directly using the system.

*2 User-2: Internet shopping site manager and operator, who manage and run the site.

— **Primary user:** Operator, who uploads and displays data of goods by using a computer, or answers end-users' questions.

— **Secondary user**: Manager of buyer department, sales department, accountant, or security control of the system.

— **Indirect user**: Owner, who has and runs the internet shopping site.

Table I.2 shows some of the important quality requirements for certain stakeholders and some of their tasks.

Table I.2 — Example of stakeholder-target matrix for other stakeholders on the case of internet shopping

Target entity		Other stakeholder				
		Products designer	*Tester*	*Manufacturer people*	*People responsible for goods transportation*	*Society*
Information system				On timeliness and accuracy of goods orders to them; **Effectiveness, Efficiency**	On timeliness and accuracy of goods orders to them; **Effectiveness, Efficiency**	Increase in unnecessary purchase of products due to ease of trans-acting [buy/sell] leading to producing extra waste/garbage; **Freedom from risks**
ICT product		To meet the business goals on the product; **Functional correctness,** **User error protection**	When pre-paring test environment; **Analyzability, Testability**			
	Technical	To meet the business needs on the product lifecy-cle; **Reusabil-ity, Modularity**				

Other stakeholder:

— Products designer

— Tester

— Manufacturer people

— People responsible for goods transportation

— Society

Annex J
(informative)

Examples of level of quality required for different ICT products(using decision table format)

Table J.1 provides an example of a decision table which shows different ICT products require different levels of quality.

Table J.1 — Examples of level of quality required for different ICT products

Condition (Classification axis)			CASE1			CASE2	CASE3
			Banking system			Meteor-ological Satellite	Mobile phone for disabilities
First layer	Second layer	Third layer	Application Processing	Information Processing	ATM		
Architecture/ Structure	Deployment structure	Hardware/ Execution environ-ment	Non-embed-ded	Non-embed-ded	Embedded	Embedded	Embedded
		System hierarchy	Information	Information	Information	Software	Computer system
			System	System	System		
		Network transpar-ency	Fixed site	Fixed site	Fixed node	Fixed node	Floating
Property	Function	Principal function	Transaction processing	Information processing	Information terminal	Equipment control	Communica-tion
	Type of information processing	Problem frame	Required-be-havior	Information Display	Command-ed-behavior	Re-quired-be-havior	Command-ed-behavior
		Style of computing	Distributed	Client-server	Client-server	Stand-alone	Stand-alone
	Size	Function size	Very large	Very large	Medium	Small	Very large
Operational environment	Application domain	Industrial domain	Financial services			Space	Tele-commu-nication
	Place to use	Area to be used	Domestic/International			Domestic	International
		Mobile readiness	Non-mobile			Mobile	Mobile
	Mission criticality	Criticality level	Social envi-ronment	Corporate manage-ment	None	National safety	None
	Aspect of provision/ acquisition	Type of provision/ acquisition	Custom-made	Cus-tom-made	Custom-made	Cus-tom-made	Embedded in commercial goods
Data	Media	Type of media	Text & numerical value			Text & numerical value	Multimedia
	Volume	Volume of data	Big data	Big data	Non-big data	Non-big data	Non-big data
	Criticality	Criticality of data	Very critical	Critical	Critical	non-critical	non-critical

Table J.1 *(continued)*

Condition (Classification axis)			CASE1			CASE2	CASE3
			Banking system			Meteor-ological Satellite	Mobile phone for disabilities
First layer	Second layer	Third layer	Application Processing	Information Processing	ATM		
Stakeholder of target system	Context of use	Type of use	Business			Business	Internet& communica-tion
	Property of users	Specificity of users		for specified users	for general users		for general users
		Number of users		Many	Myriad		Myriad
		Degree of user's proficiency		for experts	for novices		for novices
		Disability		for non-disa-bled	for disabled		for disabled
	Type of interaction	Interactiveness	Non-interac-tive	Interactive	Interactive	Non-interac-tive	Interactive
Action (im-portance of quality char-acteristics) H: High M: Middle L: Low N: Not re-quired	Functional suitability	Functional com-pleteness	H	H	H	H	M
		Functional cor-rectness	H	H	H	H	M
		Functional appro-priateness	H	H	H	H	M
	Reliability	Maturity	H	H	H	H	M
		Availability	H	M	M	L	L
		Fault tolerance	H	M	L	H	L
		Recoverability	H	H	H	H	H
	Perfor-mance efficiency	Time- behavior	H	M	M	H	H
		Resource utilization	M	L	M	H	H
	Usability	Appropriateness recognizability	N	H	M	N	H
		Learnability	N	M	M	N	H
		Operability	N	M	H	N	H
		User error protec-tion	N	H	H	N	H
		User interface aesthetics	N	L	L	N	H
		Accessibility	N	L	H	N	H
	Security	Confidentiality	H	H	H	L	H
		Integrity	H	H	H	H	H
		Non-repudiation	H	H	H	L	L
		Accountability	H	H	H	L	L
		Authenticity	H	H	H	H	H
	Compatibil-ity	Co-existence	L	L	L	L	H
		Interoperability	L	L	L	L	H

Table J.1 *(continued)*

Condition (Classification axis)			CASE1			CASE2	CASE3
			Banking system			Meteor-ological Satellite	Mobile phone for disabilities
First layer	Second layer	Third layer	Application Processing	Information Processing	ATM		
	Maintaina-bility	Modularity	H	H	H	L	H
		Reusability	L	L	H	M	H
		Analyzability	H	H	M	H	H
		Modifiability	H	H	H	H	H
		Testability	H	H	H	H	H
	Portability	Adaptability	M	M	M	L	H
		Installability	H	H	H	L	M
		Replaceability	M	M	H	L	L

NOTE 1 A system can be composed of several subsystems, each of which has different properties and therefore different quality characteristics.

NOTE 2 The classification axes used in this table are selected from ISO/IEC TR 12182.

Annex K
(informative)

IT service quality requirements

ISO/IEC TS 25011 IT service quality model has been published in the SQuaRE series; this annex describes how IT service quality requirements are to be treated.

The scope of the IT service quality related requirements is shown in Figure K.1, which is made by interpreting ISO/IEC TS 25011:2017, Figure 5.

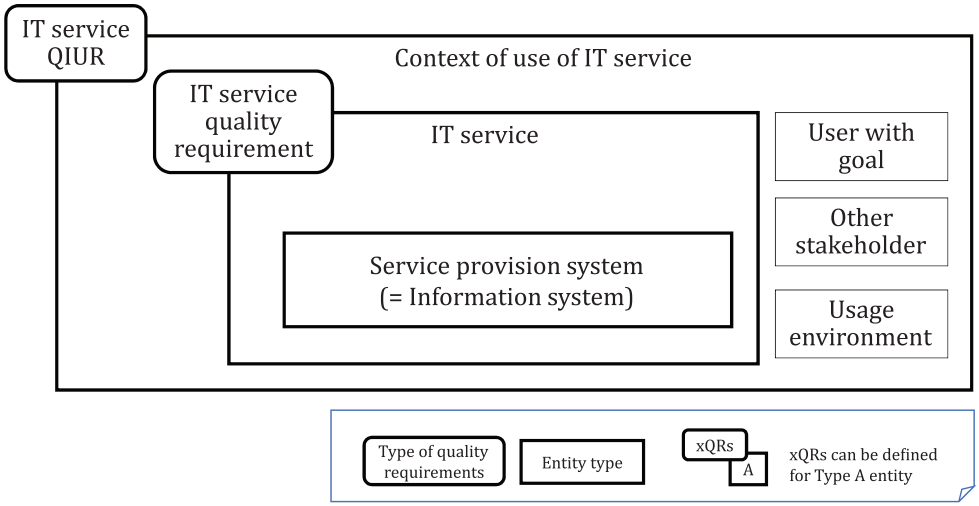

Figure K.1 — Scope of IT service related requirements and relationship to information system

In Figure K.1, the scope of the IT service quality requirements is IT service, and the scope of QIURs (defined in the IT service context) is context of use of the IT services. The target IT service is provided by some service provision systems, whose quality can influence the IT service quality and the quality in use of its context of use. Therefore some quality requirements for the service provision systems are derived from IT service quality requirements for the target service and QIURs for their context of use.

Note that a service provision system is an information system, whose quality requirements can be elicited, defined, used and governed using the framework for quality requirements.

Bibliography

[1] ISO/IEC TR 12182:2015, *Systems and software engineering — Framework for categorization of IT systems and software, and guide for applying it*

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

[3] ISO/IEC 25001, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Planning and management*

[4] ISO/IEC TS 25011:2017, *Information technology — Systems and software quality requirements and evaluation (SQuaRE) — Service quality models*

[5] ISO/IEC 25020, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Measurement reference model and guide*

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

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

[8] ISO/IEC 25063, *Systems and software engineering — Systems and software product Quality Requirements and Evaluation (SQuaRE) — Common Industry Format (CIF) for usability: Context of use description*

[9] ISO 25065, *Systems and software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Common Industry Format (CIF) for Usability: User requirements specification*

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

[11] ISO/IEC/IEEE 29119-2, *Software and systems engineering — Software testing — Part 2: Test processes*

[12] ISO/IEC/IEEE 29148, *Systems and software engineering — Life cycle processes — Requirements engineering*

[13] IEEE 730:2014 — *IEEE Standard for Software Quality Assurance Processes*

ICS 35.080
Price based on 46 pages