INTERNATIONAL STANDARD

ISO/IEC 25020

Second edition
2019-07

Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality measurement framework

Ingénierie des systèmes et du logiciel — Exigences de qualité du produit logiciel et évaluation (SQuaRE) — Modèle de référence de mesure et guide

Reference number
ISO/IEC 25020:2019(E)

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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).

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

For an explanation on the 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 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 25020:2007), which has been technically revised.

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

- relationships among different types of quality measures have been added;
- application of measurement results and description of quality measure have been added;
- elements for documenting quality measures in Annex C have been supplemented and categorized;
- Annex D has been added showing a normalized measurement function for QMs;
- Annex E has been added showing the measurement information model in ISO/IEC/IEEE 15939;
- harmonized with ISO/IEC 25000:2014, ISO/IEC 25022:2016, ISO/IEC 25023:2016, ISO/IEC 25024:2015 and ISO/IEC/IEEE 15939:2017.

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 https://www.iso.org/members.html.

Introduction

0.1 General

This document is a part of the SQuaRE series of International Standards. It provides a framework for measuring the quality characteristics and sub-characteristics (defined in ISO/IEC 2501n). This document serves as a guideline for developing and selecting quality measures for quality in use (in conjunction with ISO/IEC 25022), system and software product quality (in conjunction with ISO/IEC 25023), data quality (in conjunction with ISO/IEC 25024) and IT service quality (in conjunction with ISO/IEC TS 25025[1)]).

0.2 Quality measurement division

This document is a part of ISO/IEC 2502n Quality Measurement Division of the SQuaRE series that consists of the following International Standards:

— ISO/IEC 25020 — *Quality measurement framework*: provides a framework for developing quality measurement;

— ISO/IEC 25021 — *Quality measure elements*: provides a format for specifying QMEs (Quality Measure Elements) and a few examples of QMEs that can be used to construct software quality measures;

— ISO/IEC 25022 — *Measurement of quality in use*: provides measures, including associated measurement functions for the quality characteristics in the quality in use model;

— ISO/IEC 25023 — *Measurement of system and software product quality*: provides measures, including associated measurement functions and QMEs for the quality characteristics in the product quality model;

— ISO/IEC 25024 — *Measurement of data quality*: provides measures, including associated measurement functions and QMEs for the quality characteristics in the data quality model;

— ISO/IEC TS 25025 — *Measurement of IT service quality*: provides measures for the IT service quality model.

Figure 1 shows the relationship between this document and other standards in the ISO/IEC 2502n division.

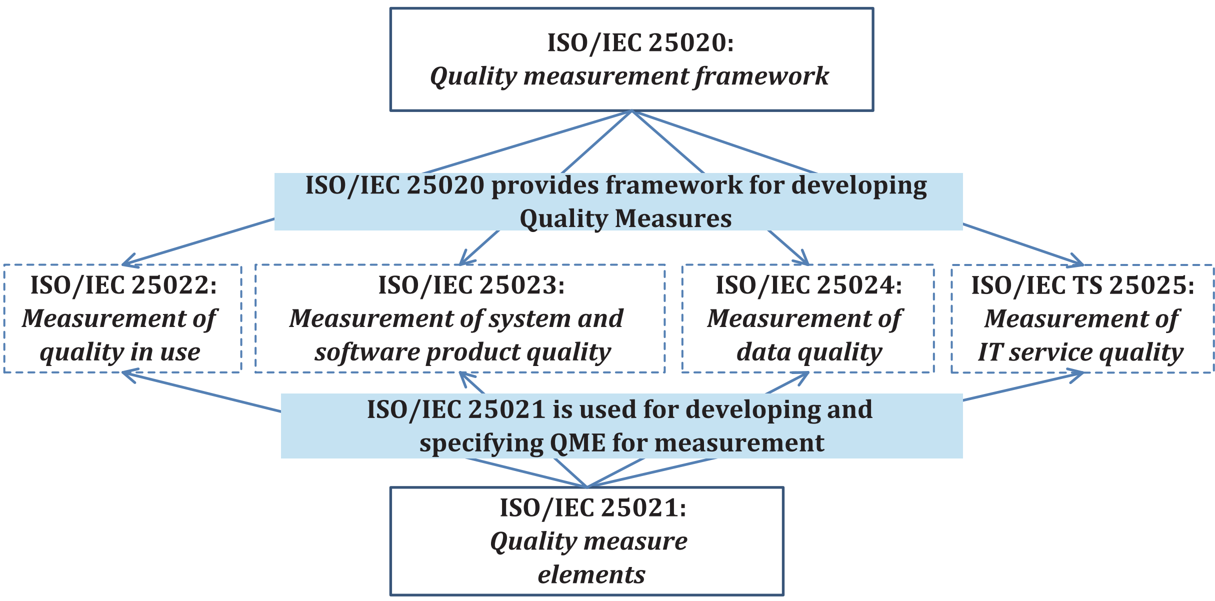


Figure 1 — Structure of the Quality Measurement Division

1) To be developed.

0.3 Outline and organization of the SQuaRE series

The SQuaRE series consists of five main divisions and one extension division. The outlines of each division within the SQuaRE series are as follows.

— **ISO/IEC 2500n - Quality Management Division**. The standards comprising this division define all common models, terms and definitions referred further 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 comprising this division present quality models for system/software products, quality in use and data. The IT service quality model is published as a Technical Specification.

— **ISO/IEC 2502n - Quality Measurement Division**. The standards comprising this division include a system/software product quality measurement reference model, definitions of quality measures, and practical guidance for their application. This division presents QMs on internal and external property of a system and software product, QMs for quality in use, QMs for data quality and QMs for IT service. Quality measure elements forming the foundations of the quality measures are defined and presented.

— **ISO/IEC 2503n - Quality Requirements Division**. The standards comprising this division help specify 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 comprising 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 quality measure as an Evaluation Module is presented as well.

— **ISO/IEC 25050-25099 - SQuaRE Extension Division.** These standards are reserved for SQuaRE extension International Standards, which currently include ISO/IEC 25051 and ISO/IEC TR 25060 to ISO/IEC 25069.

Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality measurement framework

1 Scope

This document provides a framework for developing quality measurement.

The contents of this document are as follows:

— quality measurement reference model;

— relationships among different types of quality measures;

— guidelines for selecting quality measures;

— guidelines for constructing quality measures;

— guidelines for planning and performing measurements;

— guidelines for the application of measurement results.

It includes considerations for selecting quality measures and quality measure elements (Annex A), assessing the reliability of measurement and the validity of quality measures (Annex B), elements for documenting quality measures (Annex C), normalized measurement function for quality measures (Annex D) and the measurement information model in ISO/IEC/IEEE 15939 (Annex E).

This document can be applied for designing, identifying, evaluating and executing the measurement model of system and software product quality, quality in use, data quality and IT service quality. This reference model can be used by developers, acquirers, quality assurance staff and independent evaluators—essentially by people responsible for specifying and evaluating the quality of information and communication technology (ICT) systems and services.

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/IEEE 15939, *Systems and software engineering — Measurement process*

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

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 25000 and ISO/IEC/IEEE 15939 and the following apply.

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

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

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

3.1
attribute
inherent property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means

Note 1 to entry: ISO 9000 distinguishes two types of attributes: a permanent characteristic existing inherently in something; and an assigned characteristic of a product, process or system (e.g. the price of a product, the owner of a product). The assigned characteristic is not an inherent quality characteristic of that product, process or system.

[SOURCE: ISO/IEC 25000:2014, 4.1, modified — Note 1 to entry has been removed; Note 2 to entry has become Note 1 to entry.]

3.2
base measure
measure (3.6) defined in terms of an *attribute* (3.1) and the method for quantifying it

Note 1 to entry: A base measure is functionally independent of other measures.

Note 2 to entry: Based on the definition of "base quantity" in the International Vocabulary of Metrology – Basic and General Concepts and Associated Terms, 2012.

[SOURCE: ISO/IEC/IEEE 15939:2017, 3.3]

3.3
derived measure
measure (3.6) defined as a function of two or more values of *base measures* (3.2)

Note 1 to entry: Adapted from the definition of "derived quantity" in the International Vocabulary of Metrology – Basic and General Concepts and Associated Terms, 2012.

[SOURCE: ISO/IEC/IEEE 15939:2017, 3.8]

3.4
indicator
measure (3.6) that provides an estimate or evaluation of specified *attributes* (3.1) derived from a model with respect to defined *information needs* (3.5)

[SOURCE: ISO/IEC/IEEE 15939:2017, 3.10]

3.5
information need
insight necessary to manage objectives, goals, risks and problems

[SOURCE: ISO/IEC/IEEE 15939:2017, 3.12]

3.6
measure, noun
variable to which a value is assigned as the result of *measurement* (3.8)

Note 1 to entry: The plural form "measures" is used to refer collectively to *base measures* (3.2), *derived measures* (3.3) and *indicators* (3.4).

[SOURCE: ISO/IEC/IEEE 15939:2017, 3.15]

3.7
measure, verb
make a *measurement* (3.8)

[SOURCE: ISO/IEC 25000:2014, 4.19]

3.8
measurement
set of operations having the objective of determining a value of a *measure* (3.6)

Note 1 to entry: Measurement can include assigning a qualitative category such as the language of a source program (ADA, C, JAVA, etc.).

[SOURCE: ISO/IEC 25000:2014, 4.20]

3.9
measurement function
algorithm or calculation performed to combine two or more *base measures* (3.2)

[SOURCE: ISO/IEC/IEEE 15939:2017, 3.20]

3.10
measurement method
logical sequence of operations, described generically, used in quantifying an *attribute* (3.1) with respect to a specified scale

Note 1 to entry: The type of measurement method depends on the nature of the operations used to quantify an attribute. Two types can be distinguished:

— subjective: quantification involving human judgment;

— objective: quantification based on numerical rules.

Note 2 to entry: Based on the definition of "method of measurement" in the International Vocabulary of Metrology – Basic and General Concepts and Associated Terms, 2012.

[SOURCE: ISO/IEC/IEEE 15939:2017, 3.21]

3.11
property to quantify
property of a target entity that is related to a *quality measure element* (3.14) and which can be quantified by a *measurement method* (3.10)

Note 1 to entry: A software artifact is an example of a target entity.

[SOURCE: ISO/IEC 25023:2016, 4.7]

3.12
quality in use
degree to which a product or system can be used by specific users to meet their needs to achieve specific goals with effectiveness, efficiency, freedom from risk and satisfaction in specific contexts of use

Note 1 to entry: Before the product is released, quality in use can be specified and *measured* (3.7) in a test environment designed and used exclusively by the intended users for their goals and contexts of use, e.g. User Acceptance Testing Environment.

[SOURCE: ISO/IEC 25000:2014, 4.24]

3.13
quality measure
QM
derived measure (3.3) that is defined as a *measurement function* (3.9) of two or more values of *quality measure elements* (3.14)

[SOURCE: ISO/IEC 25021:2012, 4.13, modified — The abbreviated term "QM" has been added.]

3.14
quality measure element
QME
measure (3.6) defined in terms of a property and the *measurement method* (3.10) for quantifying it, including optionally the transformation by a mathematical function

Note 1 to entry: The system or software quality characteristic or subcharacteristic of the entity is derived afterwards by calculating a software *quality measure* (3.13).

[SOURCE: ISO/IEC 25000:2014, 4.26, modified — The abbreviated term “QME” has been added.]

3.15
quality measure on external property
QM on external property
measure (3.6) of the degree to which a system or software product enables its behaviour to satisfy stated and implied needs for the system including the software to be used under specified conditions

Note 1 to entry: *Attributes* (3.1) of the behaviour can be *measured* (3.7), verified and/or validated by executing the system or software product during testing and operation.

EXAMPLE The failure density against test cases found during testing is a quality measure on external property related to the number of faults present in the computer system. The two measures are not necessarily identical since testing may not find all faults, and a fault may give rise to apparently different failures in different circumstances.

[SOURCE: ISO/IEC 25000:2014, 4.11, modified — The term has been changed from “external measure of system or software quality" to "quality measure on external property”; "QM on external property" has been added as an alternative; in Note 1 to entry, the word "measured" has been added; in EXAMPLE, "number of failures" has been changed to "failure density against test cases".]

3.16
quality measure on internal property
QM on internal property
measure (3.6) of the degree to which a set of static *attributes* (3.1) of a software product satisfies stated and implied needs for the software product to be used under specified conditions

Note 1 to entry: Static attributes include those that relate to the software architecture, structure and its components, data structure and its formats, structure and appearance of graphical display on screen and menus for users or recipients of service.

Note 2 to entry: Static attributes can be verified by review, inspection, simulation and/or automated tools.

Note 3 to entry: Quality measures on internal property are typically associated with quality requirements on static properties and attributes that can be specified in or derived from requirements.

EXAMPLE Complexity measures and the number, severity, and failure frequency of faults found in a walk through are typical quality measures on internal property made on the product itself.

[SOURCE: ISO/IEC 25000:2014, 4.16, modified — The term has been changed from “internal measure of software quality" to "quality measure on internal property"; "QM on internal property" has been added as an alternative; in Note 1 to entry, more information on static attributes has been added; Note 3 to entry has been added.]

3.17
system and software product quality
product quality
capability of a system and/or software to satisfy stated and implied needs when used under specified conditions

Note 1 to entry: Product quality model refers to the system and software product quality model defined in ISO/IEC 25010.

4 Abbreviated terms

ICT	Information and Communication Technology
QM-RM	Quality Measurement Reference Model
QM	Quality Measure
QME	Quality Measure Element

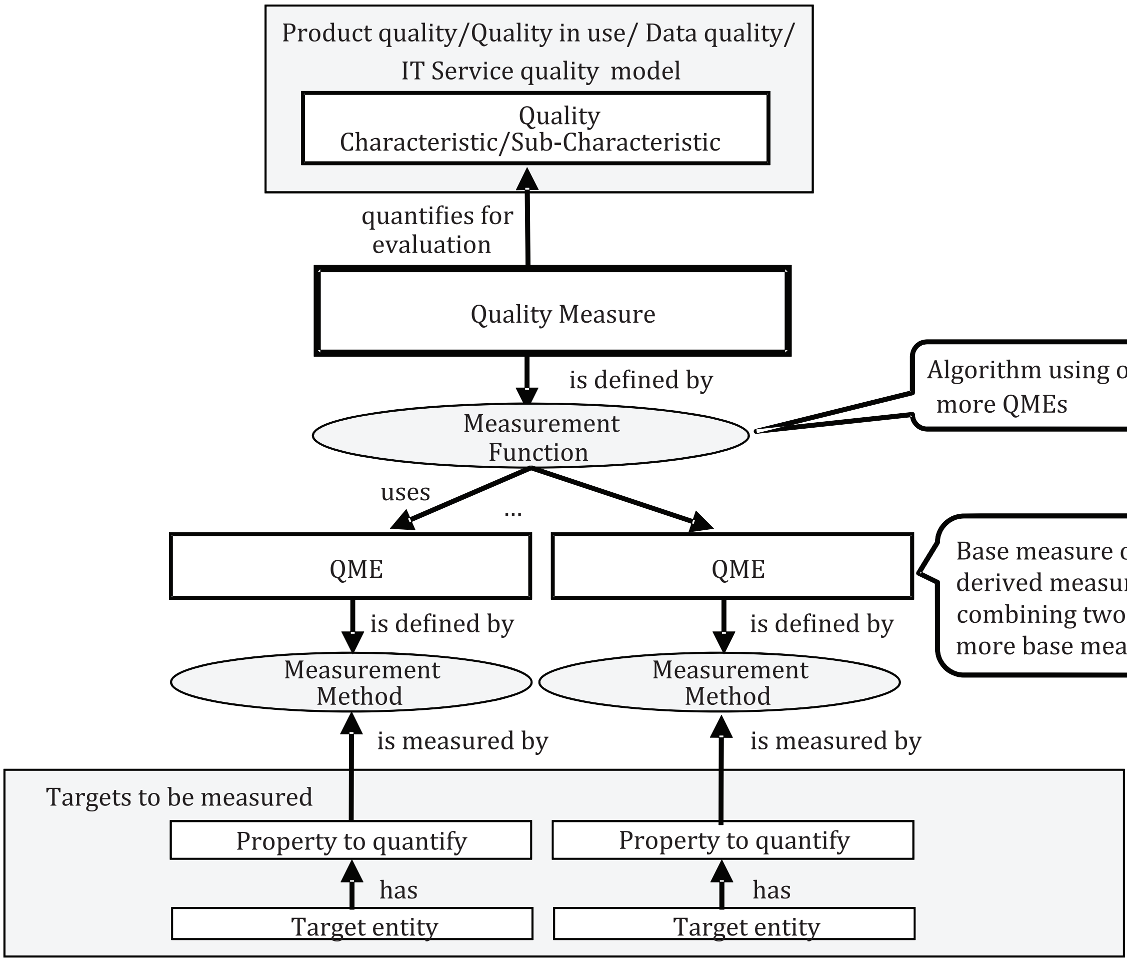
5 Conformance

Any measurement process for system and software product quality and quality in use, data quality and IT service quality that conforms to this document shall fulfil the requirements of Clause 6.

6 Quality measurement

6.1 Quality measurement reference model

The Quality Measurement Reference Model (QM-RM) describes the relationship between a quality model and the construction of QMs from QMEs, as shown in Figure 2. The relationship constitutes the reference model for the measurement of system and software product quality, quality in use, data quality and IT service quality. The measurement information model presented in Annex E describes the relationship between attributes and measurement.



NOTE Target entity can be a system, a software product, data or IT service.

Figure 2 — Quality Measurement Reference Model (QM-RM)

The quality of a system, software product, data or IT service is the degree to which it satisfies the stated and implied needs of various stakeholders, and thus provides value. User needs for quality include requirements for system quality in specific contexts of use. These stated and implied needs are represented in the SQuaRE series of standards by quality models that categorise quality into characteristics, which are further subdivided into sub-characteristics. Quality properties are measured using a measurement method. A measurement method is a logical sequence of operations used to quantify a property against a specified scale. The result of applying a measurement method is called a QME.

QMs are constructed by applying a measurement function to a set of QMEs. A measurement function is an algorithm used to combine QMEs. The result of applying a measurement function is called a QM. In this way, QMs serve as quantifications of quality characteristics (and sub-characteristics). More than one QM may be used for measuring a quality characteristic (and sub-characteristics).

In the special case where the QME serves as a QM as well, the measurement function applied would be the identity function. QMEs may either be base or derived measures. Annex B provides assessment information for the validation and verification of the measure. QMEs are constructed based on the guidance provided in ISO/IEC/IEEE 15939. Refer to ISO/IEC 25030 for guidance on selecting quality characteristics and sub-characteristics of interest in conjunction with the specification of quality requirements and ISO/IEC 25040 for guidance on using software QMs for software product evaluation.

6.2 Different QMs and their relationships

There are four quality models in the SQuaRE series:

a) the quality in use model in ISO/IEC 25010 that can be applied to software products, systems and IT services;

b) the product quality model in ISO/IEC 25010 that can be applied to systems and software products;

c) the data quality model in ISO/IEC 25012 that can be applied to data within a computer system and used by humans and systems;

d) the IT service quality model defined in ISO/IEC/TS 25011 that can be applied to IT services that support the needs of an individual user or a business.

These models provide a set of quality characteristics and sub-characteristics, as well as their definitions. The relationships among various QMs for different quality models from the SQuaRE series are shown in Figure 3.

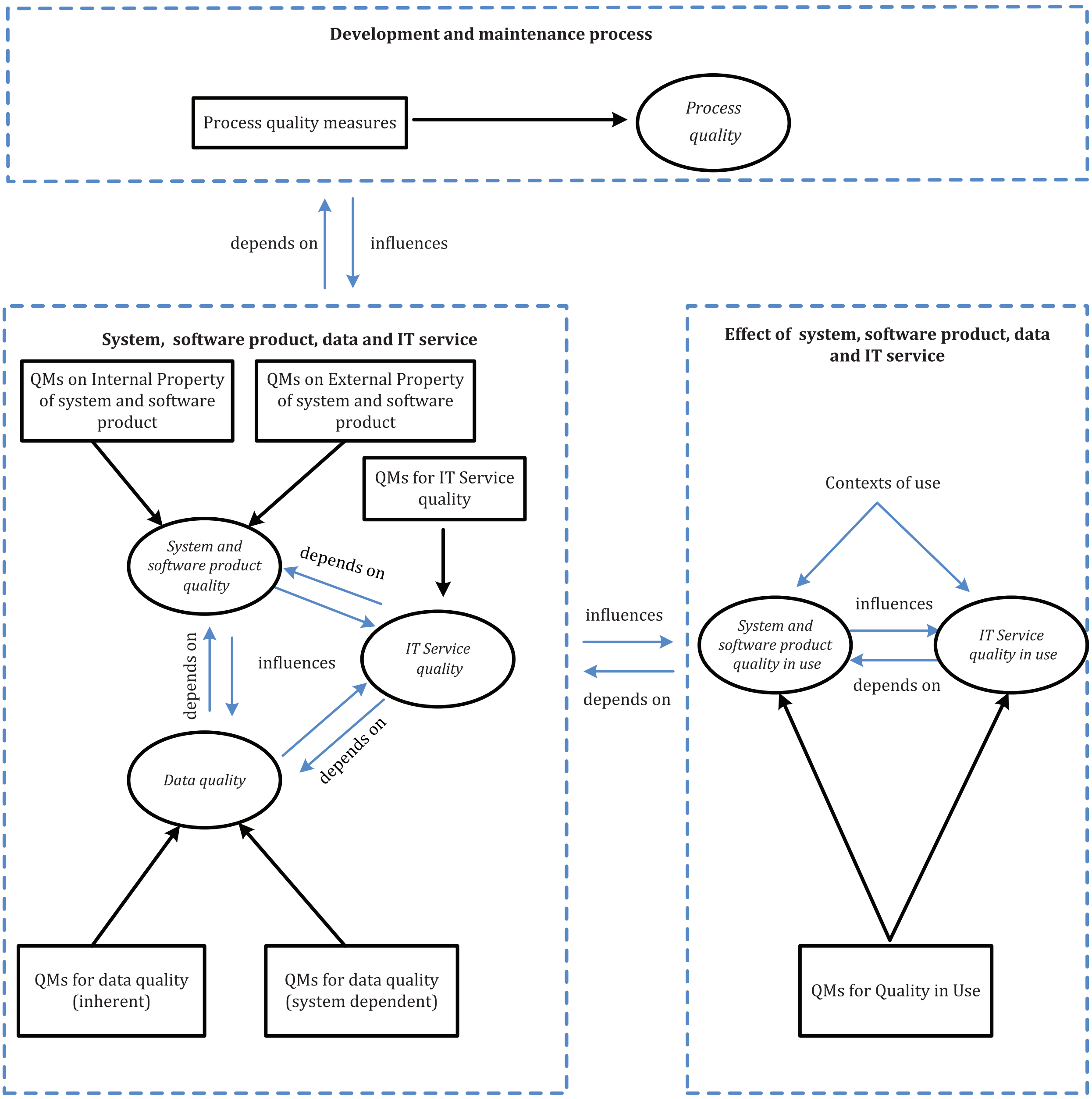


Figure 3 — Relationships among different QMs

QMs for quality in use are defined or selected to specify stakeholder requirements in a specific context of use, when quality requirements are derived from user needs by analysing the concept of operation. QMs for quality in use are to measure the extent to which a product meets the needs of specific users with respect to their specific personal or business goals by means of quantifying outcomes of interaction between a user and a system or of effects to stakeholders, including indirect users as well as direct users. These measures can only be prepared in a realistic and operational system environment.

QMs on external and internal property of product are for the user (including executing testing engineer) and the developer, respectively. There is no distinction between the two, even at the level of characteristics and sub-characteristics. However, when users apply the QMs depending on the purpose and stage of the software product life cycle, the QME and QM selected should be related and suitable to either the user or the developer. QMs on external property are used to measure the quality of the system and software product based on the behaviour of the system. QMs on external property are used in the testing and operational stages of the product life cycle. QMs on internal property allow users to measure the quality of intermediate deliverables or work products. Additionally, these measures may be used with an analysis model to predict the quality of the final system and software product. This allows users to detect system and software product quality issues and take corrective and preventive actions during the early stages of the development life cycle.

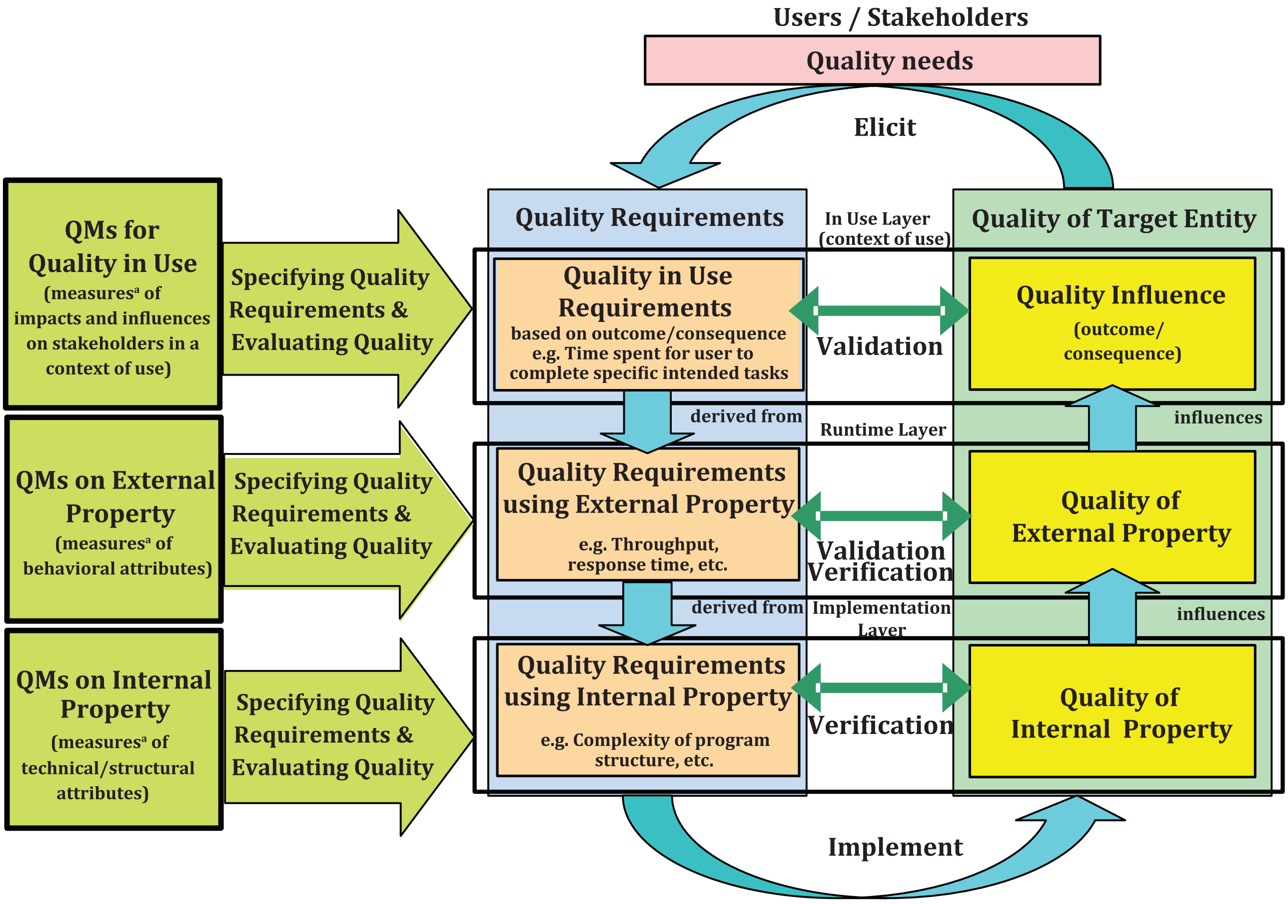
Data quality measures can be transformed from quality in use, system and software product quality requirements and measures. Then, these measures representing the targeted data quality requirements are used to evaluate the data quality of system and software product, to verify, validate and improve data and product phase-by-phase during design, implementation, testing or in use. QMs for data quality are to measure data in the system and software product from two viewpoints, namely, “Inherent” and “System-dependent”, to detect potential quality problems related to the data and database. These QMs can be applied during the development, testing and operation stages. Data quality has a big influence on quality in use, in particular for effectiveness, usefulness and risks management.

QMs for IT service quality quantify the degree to which the properties of an IT service can satisfy the stated and implied needs of the IT service when used under specified conditions. IT service has its own provision system. QMs for IT service quality typically measure interactions between the system and service recipients.

Process quality (the quality of any life cycle process defined in ISO/IEC/IEEE 12207) contributes to the improvement in system and software product quality, data quality and IT service quality. Evaluating whether software products can satisfy users’ quality needs is one of the processes in the software development life cycle. Software product, IT service and data in different contexts influence quality in use. Therefore, assessing and improving a process is a means for improving system and software product quality, and evaluating and improving system and software product quality is one means of improving quality in use. Similarly, evaluating quality in use can provide feedback for improving a software product, and evaluating a software product can provide feedback to improve a process. System and software product quality can be evaluated using QMs on internal and external property. System and software product quality influences IT service quality and data quality. IT service quality can be evaluated using IT service quality measures. IT service quality depends on system and software product quality and data quality. In the specific context of use (when system, software product and IT service are in real or simulated use), IT service quality in use depends on system and software product quality in use.

Figure 4 illustrates the quality life cycle as a set of coordinated QMs which can be used to specify quality requirements in detail and evaluate quality by means of measuring the degree of achievement of the required quality for verification and validation, through the life cycle, covering development, operation and maintenance of the system and software product, data and IT service. From user and/or stakeholder’s point of view, the quality life cycle consists of 3 layers: user layer, runtime layer and implementation layer. Quality requirements and target entity validate and/or verify each other in different layers. Users and/or stakeholder’s quality needs for any of various target entities including system, software product, data and IT service can be elicited and transformed into quality in use requirements, and then into quality requirements using external property (i.e. behaviours) and into quality requirements using internal property (i.e. static attributes). Correspondingly, target entity can be implemented from the requirements. Conducting and iterating the quality life cycle leads to evolving and improving the quality.

QMs include QMs for quality in use, QMs on external property and QMs on internal property. Stakeholders' impact and influence in a context of use can be measured by QMs for quality in use. QMs on external property are measures of behavioural attributes. QMs on internal property are used to measure technical/structural attributes of software and/or system. Quality property of target entity includes external quality property and internal quality property. Internal quality property influences external quality property when the software and/or system are in the status of runtime, while outcome or consequence of software and/or system in a certain context of use is influenced by external quality property.



a Measures are constructed using measurement functions applied.

Figure 4 — QMs in the quality life cycle

Quality in use requirements are based on the expected outcome/consequence of system and/or software product (e.g. time spent for user to complete specific intended tasks), considering effectiveness, efficiency, satisfaction, freedom from risk and context coverage. Quality requirements using external property (e.g. throughput, response time, etc.) can be derived from quality in use requirements. Quality requirements using external property should be stated quantitatively in the quality requirements specification by using QMs on external property that are used when a target entity is evaluated. Quality requirements using internal property (e.g. complexity of program structure, etc.) can be derived from quality requirements using external property. Quality requirements using internal property reflect the technical/structural property. They may be used to specify properties of deliverable, non-executable software products such as documentation and manuals. They can also be used as target entity for verification, and to define the criteria for verification at various stages of development.

Throughout the quality life cycle, measuring the degree of achievement to the required quality for verification and validation can be done in different layers. In use layer, the context of use plays an important role in validation between quality in use requirements and quality influence. In runtime layer, the quality requirements using external property are validated and verified based on quality of external property, and vice versa. In implementation layer, the quality requirements using internal property are verified based on quality of internal property, and vice versa.

NOTE QMs for quality in use indicate quality explained by the effect to stakeholders; QMs on external property indicate quality explained by the behaviour of the target entity, during prototyping test, product test, and when actually used; QMs on internal property indicate quality explained as a result of reviewing specifications and/or source code.

6.3 Selecting QMs

QMs are specified to satisfy the needs of developers, acquirers, managers, direct and indirect users and other stakeholders for information. Candidate QMs that potentially satisfy quality requirements should be identified from standards providing specific QMs in Quality measurement division of SQuaRE series, such as ISO/IEC 25022, ISO/IEC 25023 or ISO/IEC 25024. And then, candidate QMs can be further refined as applicable. At a minimum, one of the criteria for selecting QMs, including their employing combinations of measures, shall be stated that provides the reason why the selected QMs are chosen. Many different combinations of base measures and derived measures may be selected to construct additional QMs, that act as indicators or address specific quality requirements. The following factors are to be considered when selecting among the alternatives: relevance to the prioritised information needs; feasibility of collecting data in the organizational unit; availability of human resources to collect and manage data; and ease of data collection. When QMs are used to define quality requirements, the criticality of such quality requirements or risk of their insufficiency can be considered as one of the criteria for selecting QMs. When QMs are used to evaluate quality, applicable rigorousness and timing can be considered as criteria.

The measures selected will influence human behavior. In many cases the behavior may lead to dysfunctional outcomes, from individuals trying to “game the system”. Users should anticipate such challenges and take actions to mitigate risks, including training, mentoring and additional strategic governance.

Criteria for selecting QMs to fulfil those information needs shall be documented.

Annex A provides examples of criteria for selecting QMs.

NOTE ISO/IEC 25030 and ISO/IEC 25040 provide guidance on quality requirements specifications and product quality evaluations, respectively.

When using a modified or a new measure that is not identified in all specific quality measurement standards, such as ISO/IEC 25022, ISO/IEC 25023 or ISO/IEC 25024, the user shall specify how the measure relates to its corresponding quality model and how it is constructed from QMEs.

Annex C provides an example of how to document a QM.

NOTE ISO/IEC 25010 provides guidance on defining and using a system and software product quality model.

6.4 Constructing QMs

6.4.1 Identify QMs needed to be constructed

The quality of a system is the degree to which the system satisfies the stated and implied needs of its various stakeholders and thus provides value. These stated and implied needs are represented in the SQuaRE series of International Standards by quality models that categorize quality into characteristics, which, in a few cases, are further subdivided into sub-characteristics. The full set of quality characteristics across these models will not be relevant to every stakeholder. Nonetheless, each category of stakeholder shall be represented in reviewing and considering the relevance of the quality characteristics in each model before finalising the set of quality characteristics that will be used, for example, to establish software product and system performance requirements or evaluation criteria.

Applicable QMs are not limited to those listed in ISO/IEC 25022, ISO/IEC 25023 and ISO/IEC 25024. If needed, a new QM may be constructed and included in the QM set of a specific characteristic or sub-characteristic to satisfy a user's additional quality requirements. The new QM should be described according to 6.4.2, and appropriate QMEs should be selected and combined using the measurement function (See Annex D).

Definitions of any new QMs, including QMs in ISO/IEC 2502n that are modified, shall be documented.

The definition of the QM should contain information included in the example format provided in Annex C.

NOTE 1 A suggested set of quality in use measures along with their definitions is given in ISO/IEC 25022.

NOTE 2 A suggested set of system and software product quality measures along with their definitions is given in ISO/IEC 25023.

NOTE 3 A suggested set of data quality measures along with their definitions is given in ISO/IEC 25024.

6.4.2 Description of the QM

The following information is important to document the definition of each QM, when the user performs the measurement of system, software product, data and IT service. The user should document additional detailed information, when describing the QM for more operational. Such more detailed information of the QM is provided in Annex C.

a) ID: Identification code of the QM. Each ID consists of the following three parts:

   — abbreviated alphabetic representing quality characteristics and possibly sub-characteristics. (for example, "PTb" denotes "Time behaviour" which measures for "Performance efficiency", "Acc" denotes measures for accuracy);

   — serial number of sequential order within the quality sub-characteristic.

   — usage tag:

      — G: generally applicable, could be used in a wide range of situations;

      — S: specialised for specific needs.

   NOTE The ID can include additional parts (e.g. PTb-1-G-IT-1 identifies a modification of PTb-1-G).

b) Name: Name of the QM.

c) Description: The information including the information needed (purpose of the measure) and quality characteristic/sub-characteristic provided by the QM and (when useful) the purpose of the measure.

d) Measurement function: Formula showing how the QMEs are combined to produce the QM.

e) Measurement method: The type of method that can be used to obtain the measure.

6.4.3 Definitions of the QMEs

QMEs are used throughout the ICT system life cycle to construct QMs of system and software product quality, quality in use, data quality and IT service quality by applying measurement methods to specified attributes and, when necessary, the measures combining QMEs via a measurement function shall be documented. The QMEs are used to measure the attributes of the system and software product itself, effects of using the system and software product in a specific context and the resources consumed or activities performed during system and software product development, testing and maintenance.

NOTE 1 The ICT system is a system that uses information and communication technologies.

NOTE 2 A suggested set of QMEs along with their definitions is given in ISO/IEC 25021.

6.5 Plan and perform measurement

The user of the ISO/IEC 2502n division of standards shall plan and perform measurement to determine the values of QMEs and QMs following the reference model in Figure 2.

The quality measurement shall be scheduled considering resources such as personnel, measurement automation, software and hardware environments. The measurement plan should contain no duplicated tasks to take the same measures to address different information needs.

NOTE 1 Some of QMEs and QMs are often planned to be taken repeatedly, iteratively or periodically to monitor trends or improvements of quality, during specific stage or along with the product life cycle.

The criteria for selecting QMs and QMEs should be considered in the measurement plan to decrease the risk of errors and reduce the planned effort, considering at least the following:

a) measurement budget;

b) the priority and strictness of QMs and QMEs that reflect critical quality requirements;

c) schedule and resources involved;

d) application of measurement result;

e) the relevance and importance of QMs based on the quality requirements and context of use.

NOTE 2 The above concerns in an individual project are often resolved by coordinating and sharing with an organizational measurement strategy providing trainings, tools, environments, personnel and so on for measurement and analysis.

The main activities associated with performing measurement are as follows:

a) identify the quality model according to the different information needs relating to quality characteristics of systems or software product, IT service, data or quality in use;

b) identify candidate and select QMs with QMEs to construct these QMs, for the identified quality model;

c) communicate measurer or data provider to take adequately QMs and QMEs, plan and collaborate with relevant stakeholders to capture QMs and QMEs;

d) generate the values of QMEs by using the measurement method;

e) calculate the value of QMs by using the measurement function;

f) verify and store the values of QMs and their QMEs with context information of measurement;

   NOTE 3 Verification of the values of QMs and QMEs can be done using various techniques, for example, range and type of values, singular values, classification errors, or large fluctuation.

g) measure quality characteristics and/or sub-characteristics by using QMs;

h) record results and inform the users who need information relating to quality for decision makings during the project or operation.

Users of the ISO/IEC 2502n division of standards for quality measurement are encouraged to comply with the quality evaluation procedures contained in the ISO/IEC 2504n division of standards and the quality requirements definition contained in the ISO/IEC 2503n division of standards.

6.6 Application of the measurement results

The measurement results can be interpreted according to quality requirements, which include system and software product quality requirement, quality in use requirement, data quality requirement and IT service quality requirement. Quality requirements are defined by using quality models and quality measures. Detailed information about the relationships among quality models, and among quality requirements, is provided in ISO/IEC 25030 respectively.

The measurement results provide information for quality evaluation. Rigorous measurements are required to make reliable comparisons between systems, software products, data and IT service respectively. Moreover, it is also required to compare the measurement results with criterion values. The measurement procedure should measure the quality characteristics (or sub-characteristics) they claim to be measuring with sufficient accuracy. Quality evaluation requirements should be allocated to suitable components to which they are related in such a way that it is possible to define each appropriate quality measure used to evaluate quality. Decision criteria should be defined for the selected individual measures. The selected quality measures should be applied to the object of evaluation according to the evaluation plan, resulting in values on the measurement scales. General requirements for the specification and evaluation of software quality are provided in ISO/IEC 25040.

A few QMs can be difficult to interpret in isolation. The following are the ways in which QMs can be applied so that they are easier to understand and interpret:

a) Conformance: comparing the measurement result with a specific business or usage requirement

   EXAMPLE 1 The maximum acceptable response time is 10 min in a specific business or usage requirement.

b) Benchmarks: comparing the measurement result with a benchmark for the same or a similar product or system used for the same purpose

   EXAMPLE 2 It is possible to complete tasks with the new system in no more time than it took with the old system.

c) Time series: comparing the measurement result over time and analysing trends

   EXAMPLE 3 The reduced number of errors made by users with each new prototype version of a system.

d) Proficiency: comparing the measurement result with the values obtained when used by a trained or expert user

   EXAMPLE 4 How much longer does it take a new user compared with an experienced user?

e) Population norms for satisfaction: when there is a database of previous values, the measurement result can be expressed as the percentage of users who have previous given a rating of at least this value. This is more suitable for the interpretation of measures of quality in use.

NOTE Measurement interpreter(s) draw some initial conclusions based on the results. However, if they are not directly involved in the technical and management processes, such conclusions should be reviewed by other stakeholders who are. All interpreters are encouraged to consider the context of the measures. For example, the interpreter(s) can be analyst(s), measurer(s), user(s) of a system, project manager(s), quality engineer(s), developer(s) and tester(s). When these interpreters belong to an acquiring or an evaluating organization that is independent from the development or maintenance, it is very important to consider the context during interpretation and to review the initial conclusion from the interpretation.

Annex A
(informative)

Considerations for selecting QMs and QMEs

A.1 Criteria for selecting QMs and QMEs

Many different combinations of quality measure elements and software quality measures may be specified to address a specific information need by a user of the ISO/IEC 2502n division of International Standards for system and software product quality measurement. The following criteria are suggested for consideration:

— relevance to prioritized quality requirements;

— ability to address all relevant quality characteristics and sub-characteristics;

— repeatability and reproducibility of the measurement;

— validity of the QM;

— feasibility of collecting data in the organizational unit;

— availability of human resources to collect, analyse and manage data;

— ease of data collection;

— availability of appropriate tools;

— privacy protection;

— ease of interpretation by the user of the measurement result;

— applicability to the context of use and/or life cycle stage evidence (internal or external to the organizational unit) of the measure’s fitness for purpose;

The costs of collecting, managing, and analysing the data at all levels should also be considered. Costs include the following:

— Measures utilization costs: associated with each measure are the costs of collecting data, automating the calculation of the measure values (when possible), analysing the data, interpreting the analysis results, and communicating the information products;

— Process change costs: the set of measures may imply a change in the development process, for example, through the need for new data acquisition;

— Special equipment: system, hardware, or software tools may have to be located, evaluated, purchased,

— Training: the quality management/control organization or the entire development team may need adapted or developed to implement the measures; and training in the use of the measures and data collection procedures. If the implementation of measures causes changes in the development process, the changes need to be communicated to the staff.

NOTE Some of the criteria are selected from the ISO/IEC/IEEE 15939 of which some are modified.

A.2 Issues affecting the reliability of measurement and the validity of QMs

A.2.1 Issues affecting the reliability of measurement

The following issues may affect the measurement reliability when applying QMEs:

a) Procedures and instruments used for collecting data

- automatically with tools or facilities/manually collection/questionnaires or interviews.

b) Quality of data

- perspective of or bias in the data (e.g. developers' self-reports, reviewers' reports, evaluators' reports);
- skills and abilities of those performing data collection (e.g. proper sampling, selecting relevant data).

A.2.2 Issues affecting the validity of QMs

QMEs and the associated measurement function used to produce QMs may affect the validity of QMs:

- measurement reliability of the QMEs used to construct the QM;
- QMEs sharing strong correlations with measures of other quality characteristics may confound the interpretation of the concerned/desired QM.

Annex B
(informative)

Assessing the reliability of measurement and the validity of QMs

B.1 Assessing the validity of QMs

B.1.1 General

Methods for demonstrating the validity of measures typically involve both a logical argument and a statistical evidence. Face validity is one type of validity. Face validity is based on a logical argument or assertion that a measure is valid. The number of failures per unit time to represent software reliability has face validity because it is logically related to the purported underlying concept. In many instances, simply documenting the rationale for the validity of a measure may be sufficient to ensure that the measure will yield meaningful results.

Statistical evidences of validity can take several forms. However, they all tend to share the idea that there is systematic variation of the measure with a known standard, be it another measure or a hypothesized reference set of values. A few examples of systematic variation are described below.

Face validity is mentioned for didactic purposes. Owing to its limited technical soundness, care should be exercised when it is used in real professional scenarios.

NOTE Validity is the degree to which an indicator measures what it is intended to measure.

B.1.2 Content validity

Content validity describes the extent to which the QMEs included in a measurement function to create a quality measure cover the domain of content referenced in the definition of the QM.

B.1.3 Construct validity

Construct validity describes the extent to which the measurement function and its associated measure elements can be demonstrated to measure the concept described in the definition of the QM.

B.1.4 Correlation

The square of the correlation coefficient indicates the percentage of variation in the values of the quality characteristics (the results of principal measures in operational use) explained by variation in the values of a quality measure.

NOTE A measurement user can predict quality characteristics without measuring them directly by using correlated measures.

B.1.5 Order preserving relationship over time

If a measure M is directly related to a quality characteristic value Q (the results of principal measures in operational use) for a given product, a change from value Q(T1) to Q(T2) would be accompanied by a change in measure value from M(T1) to M(T2) in the same direction (for instance, if Q increases, M increases).

NOTE A measurement user can detect the movement of quality characteristics along a period without measuring directly by using those measures that have tracking ability.

B.1.6 Order preserving relationship across products

If quality characteristics values (the results of principal measures in operational use) Q1, Q2,…, Qn, corresponding to products 1, 2,…, n, have the relationship Q1 > Q2 > …> Qn, the corresponding measure values would have the relationship M1 > M2 > …> Mn. This is an important form of the statistical evidence of reliability.

NOTE A measurement user can notice exceptional and error-prone components of software by using those measures that are capable of being consistent.

B.1.7 Predictive validity

If a measure is used at time T1 to predict a quality characteristic value Q (the results of principal measures in operational use) at time T2, the prediction error, which is {[predicted Q(T2) − actual Q(T2)] / actual Q(T2)}, would be within the allowed prediction error range.

NOTE A measurement user can predict the movement of quality characteristics in the future by using those measures that are within the allowed prediction error range.

B.1.8 Discrimination

A measure should be able to discriminate between high and low quality for software characteristics and sub-characteristics.

NOTE A measurement user can categorize software components and rate quality characteristic values by using those measures that can be used to discriminate between high and low quality.

B.2 Assessing the reliability of measurement

The measurement reliability is most important with respect to collecting base measures. Methods for establishing the reliability of a measure typically involve taking repeated measurements under the same or similar conditions and assessing the variation in those measurements. The relevant conditions include the instrumentation to collect the measure such as automation, survey, human counting or human judgement, as well as the conditions within which the instrumentation is applied. In the SQuaRE series, the measurement reliability is primarily a concern for the selection and collection of the QMEs defined in ISO/IEC 25021. As noted in ISO/IEC/IEEE 15939:2017, Annex D, the reliability of a measurement method should be approached from two perspectives:

— Repeatability: the degree to which repeated use of the base measure in the same organizational unit following the same measurement method under the same conditions (e.g. tools, individuals performing the measurement) produces results that can be accepted as being identical.

— Reproducibility: the degree to which repeated use of the measure in the same organizational unit following the same measurement method under different conditions (e.g. tools, individuals performing the measurement) produces results that can be accepted as being identical.

Repeatability characterizes the extent of variation inherent in a single measurement method. Reproducibility characterizes the amount of variation in the measures due to other sources such as choice of tools, extent of training and personal differences. Various statistics have been developed to characterize the measurement reliability. For measures using an ordinal or nominal scale, the Kappa statistic can be used. For measures using an interval or absolute scale, Cohen’s alpha or other correlation-based measures can be used. More information about the measurement reliability can be found within the field of measurement system evaluation.

NOTE Reliability is the degree to which a measure repeatedly and consistently produces the same result.

Annex C
(informative)

Elements for documenting QMs

Table C.1 provides elements for documenting the QMs. The ITEM column indicates the recommended content for a system and software product quality measure definition. The CONTENT column describes what should be included in this field, as well as suggestions about where to find content within the SQuaRE series of standards. The MANDATORY column shows whether the item is mandatory or optional.

Table C.1 — Elements for documenting QMs

ITEM	CONTENT	MANDATORY (Y/N)
ID	Identification code of the QM. Each ID consists of the following three parts: — abbreviated alphabetic representing quality characteristics and possibly sub-characteristics. (for example, “PTb” denotes “Time behaviour” and measures for “Performance efficiency”, “Acc” denotes measures for accuracy); — serial number of sequential order within the quality sub-characteristic. — usage tag: — G: generally applicable, could be used in a wide range of situations; — S: specialised for specific needs. NOTE The ID can include additional parts (e.g. PTb-1-G-IT-1 identifies a modification of PTb-1-G).	Y
System and software quality measure name	Assigned name of the QM. This is taken from ISO/IEC 25022 through ISO/IEC 25024 or is provided by the user. EXAMPLE Estimated latent fault density.	Y
System and software product quality characteristic	Quality characteristic from the quality model used. This is taken from ISO/IEC 25022 through ISO/IEC 25024 or is provided by the user based on the quality model being used. EXAMPLE System and software product quality–reliability.	Y
Sub-characteristic	Quality sub-characteristic, if applicable. This is taken from ISO/IEC 25022 through ISO/IEC 25024 or is provided by the user based on the quality model being used. EXAMPLE System and software product quality–maturity.	Y
Measurement focus	Applicable portion of product quality life cycle; QM on internal property, QM on external property, or quality in use. These correspond to product quality life cycle phases, as described in ISO/IEC 25010. If a user is using a different system and software product quality model, the user should provide this information as applicable. EXAMPLE System and software products quality (testing phase).	Y

Table C.1 *(continued)*

ITEM	CONTENT	MANDATORY (Y/N)
Description of system and software quality measure (information need)	Should be a declarative statement. Often the purpose of the QM will be for evaluation against criteria established as part of the definition of a quality requirement. A specific question that the quality measure answers may also be included as part of the purpose. The following may be used as a template for this statement: <Verb> the <object of interest> to <statement of why make the measurement>. EXAMPLE Evaluate code quality by monitoring the test process and the resulting fault density to determine the probability of satisfying the reliability requirements. Question: How many future faults might we find?	Y
Decision criteria	Decision criteria are numerical thresholds or targets used to determine the need for action or further investigation, or to describe the level of confidence in a given result. These are often set with respect to quality requirements and the corresponding evaluation criteria. Moreover, users may use benchmarks, statistical control limits, historical data, customer requirements or other techniques to set decision criteria. If this information is documented elsewhere, a reference to that location is adequate. EXAMPLE If the estimated defect density exceeds the acceptable threshold, perform additional defect detection and removal activities.	N
Measurement function	Equations showing how quality measure elements are combined to produce a quality measure. EXAMPLE Estimated latent defect density = (C1–C2)/S.	Y
Quality measure elements used	Name and definition of the quality measure element used. If the quality measure element is defined elsewhere, a reference to that location is adequate. Add as many rows as are needed. See Annex A for criteria to specify QMEs. EXAMPLE C1: Total number of predicted latent faults in a system and software product. C2: Cumulative number of unique faults detected. S: Product size.	Y
Measurement method	Describe the measurement method for the QMEs. If these are described elsewhere such as in ISO/IEC 25021, a reference to that description can be provided instead of a full description. EXAMPLE C1: Predicted number of faults using historical defect density. C2: Count of defects reported in defect tracking system. S: Count of non-comment lines of code.	N
Data source(s)	Describe the data source(s) of the QMEs. If these are described elsewhere, such as in ISO/IEC 25021, a reference to that description can be provided instead of a full description. EXAMPLE C1: Organization historical database. C2: Defect tracking system. S: Software source code file in configuration management system.	N

Table C.1 *(continued)*

ITEM	CONTENT	MANDATORY (Y/N)
Evidence of validity of measures	A statement about the extent to which the QM meets this selection criterion and a description of the method and evidence used to make the determination. An ordinal scale of high, medium or low with respect to the relationship between the measure and the purpose may be used. See Annex B for information regarding the measurement validity. The following template can be used: "The validity of <measure> is <rating> dependent on <evidence of validity> …." EXAMPLE The validity of code maturity is highly dependent on the logical association between the fault density and the code maturity: the lower the fault density, the higher the assumed maturity of the code and the more reliable the code maturity.	N
Reliability evidence of measurement	A statement about the extent to which the QM satisfies this selection criterion and a description of the method and evidence used to make the determination. An ordinal scale of high, medium or low based on the measurement method and the underlying assumptions may be employed. Moreover, there are statistical methods for assessing the measurement reliability, See Annex B for additional information on methods for establishing the reliability of a measure. The following template can be used: "The reliability of <measure> is <rating> dependent on <evidence of reliability> …." EXAMPLE The reliability of defects per function point is highly dependent on counting of number of defects and functional size, furthermore, adherence to testing methods and functional size measurement standards. See ISO/IEC 14143-6, ISO/IEC 20926, ISO/IEC 19761, ISO/IEC 29881, ISO/IEC 20968, ISO/IEC 24570 for different functional size methods.	N
Cost of measurement	A statement about the extent to which the QM satisfies this selection criterion and a description of the method and evidence used to make the determination. An ordinal scale of high, medium or low based on an analysis of the costs associated with collecting the QMEs may be used. Examples of cost considerations include whether the data are already being collected, whether the collection will require new tools or be performed manually and the volume of data to be collected. EXAMPLE Low. Such tools or environments are usually available to calculate the prediction model and size measurement. (Some additional cost may be incurred if a new prediction model will be developed.)	N
Usage scenarios by role	A description of how the QM would be used to fulfil the measurement purpose. This should include who will use the measure, when they would use it and who would be affected by the various types of decisions that might be made based on the measurement results. EXAMPLES Software Quality Assurance personnel can use this QM to evaluate the estimated fault density during qualification testing. Trends in this measure can be used to evaluate the status of defect removal activities and improvement in the reliability of the software as a part the of quality assurance process. Developers or testers can use this QM to evaluate the estimated fault density during software integration testing. Trends in this measure can be used to evaluate the status of defect removal activities and improve the reliability of the software as a part of the decision to release the code for the next phase of testing.	N

Annex D
(informative)

Normalized measurement function for QMs

The range of QM values and the trends of changes in them possibly vary too widely to be displayed concisely. Such a problem can be resolved by employing examples of the measurement function shown here. The values of measure elements can be transformed to QM values ranging from 0 to 1 by using measurement functions to acquire quantitative and comparable values for evaluating characteristics and sub-characteristics.

The formulas of the measurement function are expressed below:

a) The user provides the maximum requirement, the real result is always a subset of the user's requirement. For example, the Fault Correction measure in Maturity is used to describe the proportion of detected reliability-related faults that have been corrected. In this case, Formula (D.1) is suitable for describing the measurement function. x is the number of reliability-related faults corrected in the design/coding/testing phase and R is the number of reliability-related faults detected in the design/coding/testing phase. The reliability-related faults corrected in the design/coding/testing phase always belong to the reliability-related faults detected. In this case, R is the maximum requirement. The value of x never exceeds the value of R. The following measure function will be used for the measurement in this scenario.

$$M = f(x) = \frac{x}{R} \tag{D.1}$$

where

M is the value of the QM;

x is the result value of the QME;

R is the expected value of the QME.

b) The user provides the upper bound of requirement but no lower bound of requirement. For example, the Mean Throughput Measure of Time behaviour stands for the mean number of jobs completed per unit time. The popular expression of this requirement is similar to "the throughput should be more than 100 transactions per second". The higher the throughput, the better the result calculated by the measure function. Formula (D.2) is suitable for describing the measure function in this scenario. Figure D.1 shows the measure function curve when R equals 100.

$$M = f(x) = \begin{cases} E \times \frac{x}{R} (0 \le x \le R) \\ 1 - (1 - E) \times \frac{R}{x} (x > R) \end{cases} \tag{D.2}$$

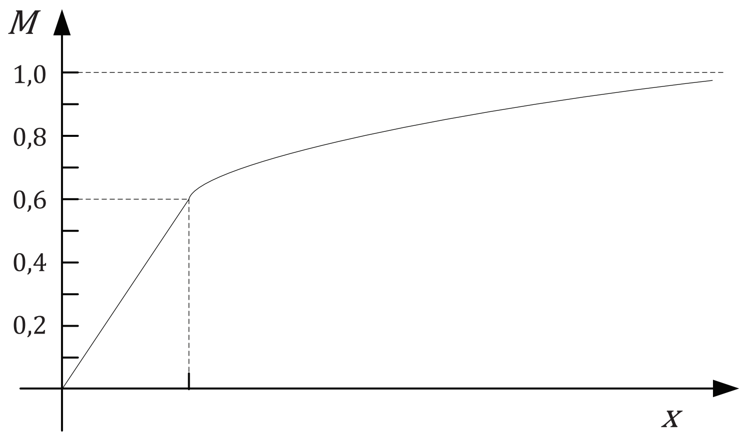
where

M is the value of the QM;

x is the result value of the QME;

R is the expected value of the QME;

E is the value of the measure index corresponding to R, decided by the user (e.g. E = 0,6).



Key

M value of the QM

x value of the QME

Figure D.1 — Relationship curve of Formula (D.2)

c) The user provides the lower bound of requirement but no upper bounder of requirement. For example, the Mean Response Time Measure of Performance Efficiency. The popular expression of this requirement is similar to “the mean response time should be less than 100 milliseconds”. The smaller the response time, the better the result calculated using the measure function. Formula (D.3) is suitable in this scenario. Figure D.2 shows the measure function curve when R equals 100.

$$M = f(x) = \begin{cases} 1-(1-E)\times\frac{x}{R}(0 \le x \le R) \\ E\times\frac{R}{x}(x > R) \end{cases} \quad \text{(D.3)}$$

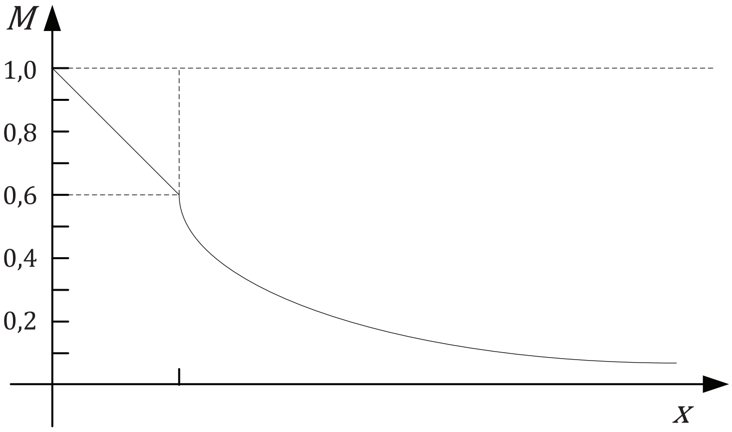
where

M is the value of the QM;

x is the result value of the QME;

R is the expected value of the QME;

E is the value of the measure index corresponding to R, decided by the user (e.g. E = 0,6).



Key

M value of the QM

x value of the QME

Figure D.2 — Relationship curve of Formula (D.3)

In different measures, *x* may have different meanings. For example, in the Functional Coverage Measure (FCp-1-G), *x* denotes the number of specified functions that has been implemented, and it equals the value of the number of functions specified minus the number of functions missing. In the Mean Down Time Measure of Availability, *x* represents the downtime per breakdown rather than the downtime.

Annex E
(informative)

Measurement information model in ISO/IEC/IEEE 15939

The measurement information model is a structure that links the information required by relevant entities and attributes of concern. For the quality discussed in this document, entities include systems, software products and data. The measurement information model describes how the relevant attributes are quantified and converted to indicators that provide a basis for decision making, as shown in Figure E.1. Detailed information about the measurement information model can be found in ISO/IEC/IEEE 15939.

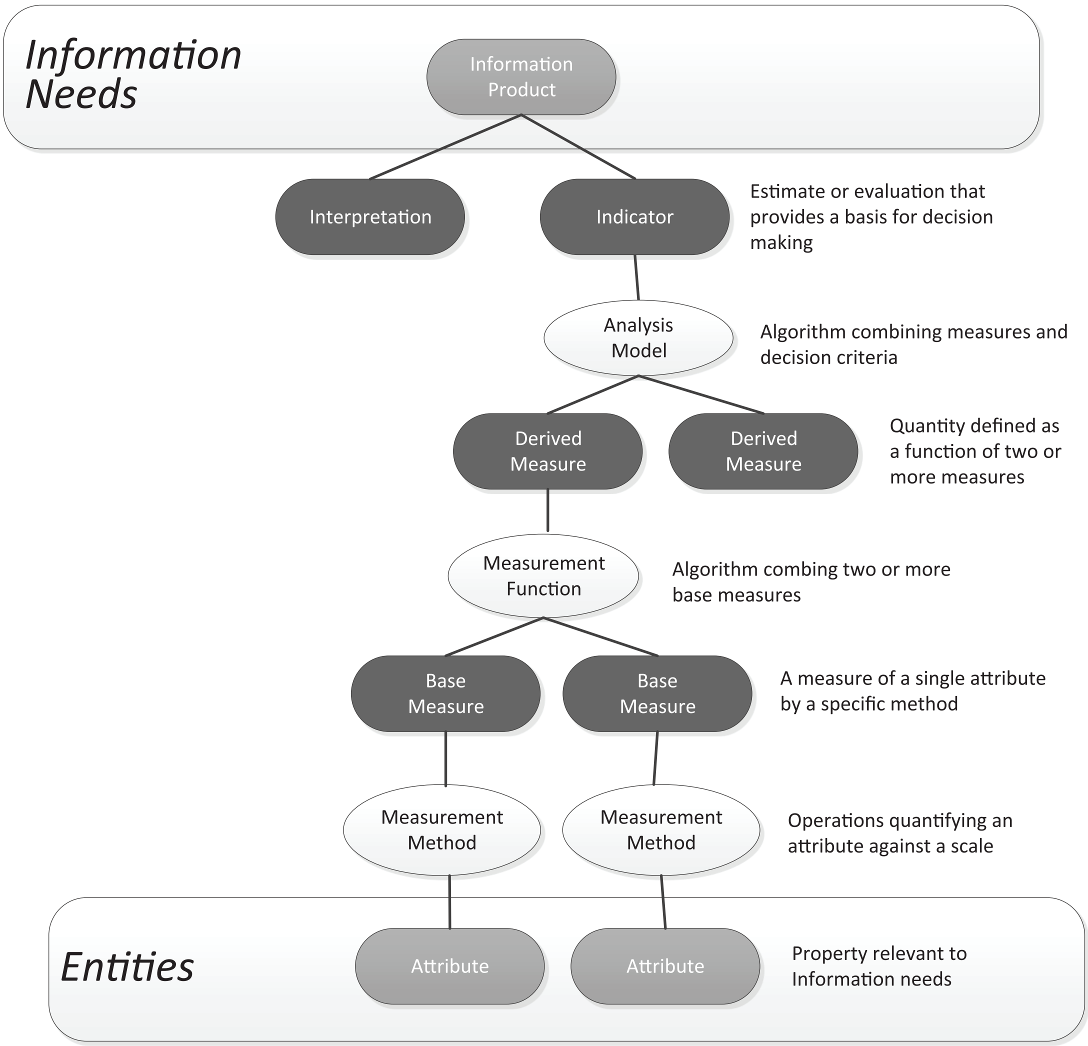


Figure E.1 — Key relationship in measurement information model in ISO/IEC/IEEE 15939

The selection or definition of appropriate measures to address an information need begins with a measurable concept: an idea of which measurable attributes are related to an information need and how they are related. The measurement planner defines measurement constructs that link these attributes to a specified information need. This measurement information model identifies basic terms and concepts. It helps determine what the measurement planner needs to specify during measurement planning, performance and evaluation.

Figure E.1 shows the relationships among the key components of the measurement information model. The model defines three types of measures: base measures, derived measures and indicators. The information content of these measures increases as they become closer in the model to the information need. Based on an understanding of the expected relationship between the component measures or their behaviours over time, a specific algorithm or calculation should be designed to combine one or more base or derived measures with associated decision criteria. A derived measure is defined as a function of two or more values of base measures. A base measure is functionally independent of other measures. It captures information about a single attribute through a measurement method. A measurement method is a logical sequence of operations, described generically, used for quantifying an attribute with respect to a specified scale. An attribute is a property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means. An entity may have many attributes, only some of which may be of interest for a measurement. Quality (sub) characteristics and their QMEs in SQuaRE can be analysed and interpreted to indicate total quality or additional quality-related matters, for example, organisational total quality or business impact, as information needs that vary in the context of measurement use. Table E.1 presents the relationship of QM-RM for SQuaRE and the measurement information model in ISO/IEC/IEEE 15939.

Table E.1 — Relationship of QM-RM for SQuaRE and measurement information model in ISO/IEC/IEEE 15939

SQuaRE	ISO/IEC/IEEE 15939
Quality of system, software product, data, IT service, quality in use or other information needs relating to quality.	**Information needs** (Insights necessary to manage objectives, goals, risks and problems [ISO/IEC/IEEE 15939:2017, 3.12])
Quality evaluation report [Based on, ISO/IEC 25040:2011, 5.2 and Annex E]	**Information product** (One or more indicators and their associated interpretations that address an information need. [ISO/IEC/IEEE 15939:2017, 3.13])
Rating level for the QM (A few QMs combine multiple QMs and QMEs and set rating levels that are used to categorise a measurement scale, for example, excellent, good, fair or poor depending on the gap between the measured value and the required value in quality requirements. [Based on ISO/IEC 25000:2014, 4.29])	**Analysis model** (Algorithm combining measures and decision criteria. [ISO/IEC/IEEE 15939:2017, Figure A.1])
Algorithm for combining QMs and decision criteria for evaluation (Procedure for the summarization with separate criteria for different quality characteristics, each of which is in terms of individual sub-characteristics and quality measures or a weighted combination of sub-characteristics and QMs. The summarization results are used to assess quality or a specific quality characteristic. [Based on ISO/IEC 25040:2011, 6.4.3])	**Analysis model** (Algorithm combining measures and decision criteria. [ISO/IEC/IEEE 15939:2017, Figure A.1])
Rating (Action of mapping the measured value to the appropriate rating level. [Based on ISO/IEC 25000:2014, 4.28])	**Interpretation** (Decision criteria help interpret the measurement results. [ISO/IEC/IEEE 15939:2017, A.2.5.1.1])
Quality evaluation based on decision criteria (Assessment of quality or a specific quality characteristic by summarization of multiple quality (sub) characteristic or QMs. [Based on ISO/IEC 25040:2011, 6.4.3])	

Table E.1 *(continued)*

<table>
<tr><th>SQuaRE</th><th>ISO/IEC/IEEE 15939</th></tr>
<tr><td>QM</td><td rowspan="3">Indicator
(Measure that provides an estimate or evaluation of the specified attributes derived from a model with respect to the defined information needs.
[ISO/IEC/IEEE 15939:2017, 3.10])</td></tr>
<tr><td>Quality (sub) characteristic</td></tr>
<tr><td>Quality</td></tr>
<tr><td>QM or QME</td><td>Derived measure
(Measure that is defined as a function of two or more values of base measures.
[ISO/IEC/IEEE 15939:2017, 3.8])</td></tr>
<tr><td>Measurement function</td><td>Measurement function
(Algorithm or calculation performed to combine two or more base measures.
[ISO/IEC/IEEE 15939:2017, 3.20])</td></tr>
<tr><td>QME</td><td>Base measure
(Measure defined in terms of an attribute and the method for quantifying it.
[ISO/IEC /IEEE 15939:2017, 3.3])</td></tr>
<tr><td>Measurement method</td><td>Measurement method
(Logical sequence of operations, described generically and used to quantify an attribute with respect to a specified scale.
[ISO/IEC/IEEE 15939:2017, 3.21])</td></tr>
</table>

Bibliography

[1] ISO 9000, *Quality management systems — Fundamentals and vocabulary*

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

[3] ISO/IEC 14143-6, *Information technology — Software measurement — Functional size measurement — Part 6: Guide for use of ISO/IEC 14143 series and related International Standard*

[4] ISO/IEC 19761, *Software engineering — COSMIC: a functional size measurement method*

[5] ISO/IEC 20926, *Software and systems engineering — Software measurement — IFPUG functional size measurement method 2009*

[6] ISO/IEC 20968, *Software engineering — Mk II Function Point Analysis — Counting Practices Manual*

[7] ISO/IEC 24570, *Software engineering — NESMA functional size measurement method — Definitions and counting guidelines for the application of function point analysis*

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

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

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

[11] ISO/IEC 25021:2012, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality measure elements*

[12] ISO/IEC 25022:2016, *Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — Measurement of quality in use*

[13] ISO/IEC 25023:2016, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality*

[14] ISO/IEC 25024:2015, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of data quality*

[15] ISO/IEC 25030, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Quality requirements*

[16] ISO/IEC 25040:2011, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation process*

[17] ISO/IEC 29881, *Information technology — Systems and software engineering — FiSMA 1.1 functional size measurement method*

[18] IEEE Std 1061-1998 (R2009), *IEEE Standard for a Software Quality Metrics Methodology, pp. 11-12, 18-19*

[19] Schneidewind Norman F., Methodology for Validating Software Metrics, *IEEE Transactions on Software Engineering*, vol. **18**, no. 5, pp. 410-422, May 1992

ICS 35.080

Price based on 27 pages