FINAL
DRAFT

INTERNATIONAL STANDARD

ISO/IEC FDIS 25020

ISO/IEC JTC **1**

Secretariat: **ANSI**

Voting begins on:
2007-01-04

Voting terminates on:
2007-03-04

Software Engineering — Software quality requirements and evaluation (SQuaRE) — Quality measurement — Measurement reference model and guide

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

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12-031
École Polytechnique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis nº 11/12-031

Please see the administrative notes on page iii

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

Reference number
ISO/IEC FDIS 25020:2007(E)

© ISO/IEC 2007

PDF disclaimer

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

Adobe is a trademark of Adobe Systems Incorporated.

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

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12-031
École Polytechique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis nº 11/12-031

Copyright notice

This ISO document is a Draft International Standard and is copyright-protected by ISO. Except as permitted under the applicable laws of the user's country, neither this ISO draft nor any extract from it may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission being secured.

Requests for permission to reproduce should be addressed to either ISO at the address below or ISO's member body in the country of the requester.

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

Reproduction may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

© ISO/IEC 2007 – All rights reserved

In accordance with the provisions of Council Resolution 21/1986, this document is **circulated in the English language only**.

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12-031
École Polytechique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis nº 11/12-031

© ISO/IEC 2007 – All rights reserved

Contents

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12-031
École Polytechnique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis nº 11/12-031

FINAL DRAFT / PROJET FINAL
© ISO/IEC 2007 – All rights reserved

Foreword

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

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

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

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 25020 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC FDIS 25020 is part of the SQuaRE series of standards and was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and Systems Engineering.*

SQuaRE series of standards consists of the following divisions under the general title *Software Engineering*, *Software Quality Requirements and Evaluation (SQuaRE):*

— Quality Management Division (ISO/IEC 2500n),

— Quality Model Division (ISO/IEC 2501n),

— Quality Measurement Division (ISO/IEC 2502n),

— Quality Requirements Division (ISO/IEC 2503n), and

— Quality Evaluation Division (ISO/IEC 2504n)

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12-031
École Polytechnique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis nº 11/12-031

© ISO/IEC 2007 – All rights reserved

Introduction

The general goal of creating the SQuaRE series of International Standards is to move to a logically organized, enriched and unified series covering three complementary processes: requirements specification, measurement and evaluation. The purpose of the SQuaRE series of International Standards is to assist those developing and acquiring software products with the specification and evaluation of quality requirements. It establishes criteria for the specification of software product quality requirements and their evaluation. It includes a two-part quality model for aligning customer definitions of quality with characteristics of the software product. In addition, the series defines measures of software product quality characteristics that can be used by developers, acquirers and evaluators.

It has to be stressed that the SQuaRE series of International Standards is dedicated to software product quality only. The Quality Management Division of the SQuaRE series deals with software products, and is separate and distinct from the "Quality Management" of processes which is defined in the ISO 9000 family of International Standards.

The major benefits of the SQuaRE series over its predecessor standards include:

— the coordination of guidance on software product quality measurement and evaluation,

— guidance for the specification of software product quality requirements, and

— harmonization with ISO/IEC 15939 in the form of Quality Measurement Reference model presented in this International Standard.

The major differences between the SQuaRE series of International Standards and its predecessors, ISO/IEC 9126 and ISO/IEC 14598, are the:

— introduction of the new general reference model,

— introduction of dedicated and detailed guides for each division,

— introduction of Quality Measure Elements within the Quality Measurement Division,

— introduction of the Quality Requirements Division,

— incorporation and revision of the evaluation processes,

— introduction of guidance for practical use in the form of examples, and

— co-ordination and harmonization of content with ISO/IEC 15939.

SQuaRE consists of the following five divisions:

— Quality Management Division (ISO/IEC 2500n),

— Quality Model Division (ISO/IEC 2501n),

— Quality Measurement Division (ISO/IEC 2502n),

— Quality Requirements Division (ISO/IEC 2503n), and

— Quality Evaluation Division (ISO/IEC 2504n).

SQuaRE provides:

— terms and definitions,

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12-031
Ecole Polytechnique de Montreal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis nº 11/12-031

FINAL DRAFT / PROJET FINAL
© ISO/IEC 2007 – All rights reserved

— reference models,

— a general guide,

— individual division guides, and

— standards for requirements specification, measurement and evaluation purposes.

SQuaRE includes International Standards and technical reports for a quality model and measures, as well as on quality requirements and evaluation. SQuaRE replaces current ISO/IEC 9126 series and ISO/IEC 14598 series.

This International Standard is intended to be used together with ISO/IEC 25010. It is strongly recommended that users refer to ISO/IEC 2500n, ISO/IEC 2501n, ISO/IEC 2503n, and ISO/IEC 2504n division of International Standards prior to using this International Standard and the associated measurement technical reports, particularly if the user is not familiar with the use of software measures for requirements specification and product evaluation. These International Standards discuss the planning and use of the software quality measures defined in the ISO/IEC 2502n series on software product quality measurement.

ISO/IEC 25010 will provide a model and defines terms for software product quality characteristics and how these characteristics are decomposed into subcharacteristics. It does not describe how any of these subcharacteristics could be measured. The Quality Measurement Division provides information and guidance about how to measure the characteristics and subcharacteristics of a quality model. This International Standard provides a reference model and guide for measuring the quality characteristics defined in ISO/IEC 2501n Quality Model Division (Figure 1). The associated standards and technical reports within the Quality Measurement Division describe measures of quality throughout the product life cycle.

ISO/IEC 25021 offers quality measure elements that can be used to construct software quality measures. Quality measure elements are the base and derived measures used to create measures of software product quality characteristics. Quality measure elements may measure a static representation of the software, the behaviour of the software, or the effects of the software when it is used.

ISO/IEC 25022, ISO/IEC 25023 and ISO/IEC 25024 will describe measures for the characteristics in the quality model. Internal measures characterize software product quality based upon static representations of the software, external measures characterize software product quality based upon the behaviour of the computer-based system including the software, and quality in use measures characterize software product quality based upon the effects of using the software in a specific context of use.

The measures in these technical reports should not be construed as an exhaustive or required set. Figure 2 depicts the relationship between this International Standard and the technical reports in the Quality Measurement Division. Developers, evaluators, quality managers, acquirers, suppliers, maintainers and other users of software may select measures from these technical reports for the measurement of quality characteristics of interest. In practice this may be with respect to defining requirements, evaluating software products, quality management and other purposes. Users may also modify the measures or use measures which are not included in those technical reports.

Provided by Standards Council of Canada
For Educational Purposes Only
Licence #12-031
Ecole Polytechnique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis n° 12-031

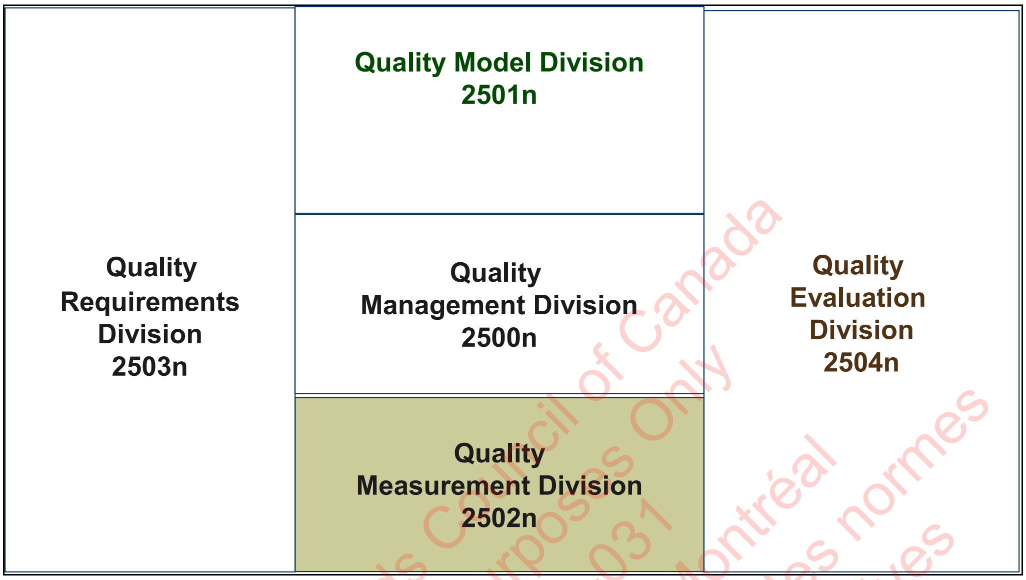

Figure 1 — Organization of the SQuaRE series of International Standards

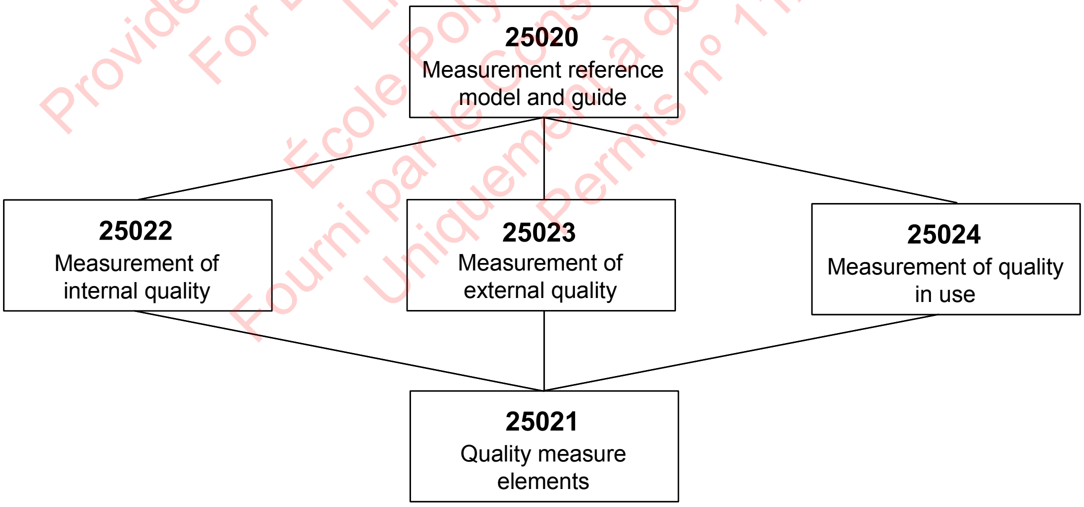

Figure 2 — Structure of the Quality Measurement division

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12-031
École Polytechnique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis nº 11/12-031

FINAL DRAFT / PROJET FINAL
© ISO/IEC 2007 – All rights reserved

Software Engineering — Software quality requirements and evaluation (SQuaRE) — Quality measurement — Measurement reference model and guide

Provided by Standards Council of Canada
For Educational Purposes Only
License #1 11/12-031
École Polytechnique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis n° 11/12-031

1 Scope

The scope of this International Standard is the selection and construction of software product quality measures, especially with respect to their use in conjunction with ISO/IEC 2503n. Software product quality requirements division and ISO/IEC 2504n. Software product quality evaluation division standards. It also provides information that assists users of the SQuaRE quality measurement technical reports (ISO/IEC 25021, ISO/IEC 25022, ISO/IEC 25023, and ISO/IEC 25024) to achieve the intended benefits of the above standards.

This International Standard also contains the following informative annexes (A through C) and Bibliography:

— Criteria for selecting software quality measures and quality measure elements

— Demonstrating predictive validity and assessing measurement reliability

— Example format for documenting software quality measures

— Bibliography

The SQuaRE series of International Standards is intended for, but not limited to, developers, acquirers and independent evaluators of software, particularly those responsible for defining software product quality requirements and for software product evaluation. It is recommended that users of SQuaRE use this International Standard as a guide to execute their software product quality measurement tasks.

2 Conformance

Any software product quality measurement that conforms to this International Standard shall fulfil the requirements of Clause 6.

3 Normative references

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

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

ISO/IEC 15939:2002, *Software Engineering — Software measurement process*

© ISO/IEC 2007 – All rights reserved

4 Terms and definitions

For the purposes of this standard, the terms and definitions given in ISO/IEC 25000 and ISO/IEC 15939 apply. The following definitions are replicated here for the convenience of the user of this Internationl Standard. Unattributed references are from ISO/IEC 25000.

4.1
attribute

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

[ISO/IEC 15939:2002]

NOTE 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.

4.2
base measure

measure defined in terms of an attribute and the method for quantifying it

NOTE A base measure is functionally independent of other measures.

[ISO/IEC 15939: 2002, based on the definition in International Vocabulary of Basic and General Terms in Metrology, 1993]

4.3
derived measure

measure that is defined as a function of two or more values of base measures

[ISO/IEC 15939:2002, based on the definition in International Vocabulary of Basic and General Terms in Metrology, 1993]

NOTE A transformation of a base measure using a mathematical function can also be considered as a derived measure.

4.4
external software quality

capability of a software product to enable the behaviour of a system to satisfy stated and implied needs when the system is used under specified conditions

NOTE Attributes of the behaviour can be verified and/or validated by executing the software product during testing and operation.

EXAMPLE The number of failures found during testing is an external software quality measure related to the number of faults present in the program. 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.

4.5
indicator

measure that provides an estimate or evaluation of specified attributes derived from a model with respect to defined information needs

[ISO/IEC 15939:2002]

NOTE In ISO/IEC 14598 this definition was, "a measure that can be used to estimate or predict another measure."

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12-031
École Polytechnique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis nº 11/12-031

FINAL DRAFT / PROJET FINAL
© ISO/IEC 2007 – All rights reserved

4.6
information need

insight necessary to manage objectives, goals, risks and problems

[ISO/IEC 15939:2002]

4.7
internal software quality

capability of a set of static attributes of a software product to satisfy stated and implied needs when the software product is used under specified conditions

NOTE 1 Static attributes include those that relate to the software architecture, structure and its components.

NOTE 2 Static attributes can be verified by review, inspection and/or automated tools.

EXAMPLE The number of lines of code, complexity measures and the number of faults found in a walk through are all internal software quality measures made on the product itself.

4.8
measure,
noun

variable to which a value is assigned as the result of measurement

NOTE The term "measures" is used to refer collectively to base measures, derived measures and indicators.

[ISO/IEC 15939:2002]

4.9
measure,
verb

make a measurement

[ISO/IEC 14598-1:1999]

4.10
measurement

set of operations having the object of determining a value of a measure

[ISO/IEC 15939:2002, based on the definition in International Vocabulary of Basic and General Terms in Metrology, 1993]

NOTE Measurement can include assigning a qualitative category such as the language of a source program (ADA, C, COBOL, etc.).

4.11
measurement function

algorithm or calculation performed to combine two or more base measures

[ISO/IEC 15939:2002]

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12-031
École Polytechnique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis nº 11/12-031

4.12
measurement method

logical sequence of operations, described generically, used in quantifying an attribute with respect to a specified scale

[ISO/IEC 15939:2002, based on the definition in International Vocabulary of Basic and General Terms in Metrology, 1993]

4.13
quality in use (measure)

extent to which a product used by specific users meets their needs to achieve specific goals with effectiveness, productivity, safety and satisfaction in specific contexts of use

4.14
quality measure element

measure, which is either a base measure or a derived measure, that is used for constructing software quality measures

NOTE The software quality characteristic or subcharacteristic of the entity is derived afterwards by calculating a software quality measure.

5 Symbols (and abbreviated terms)

The following symbols and abbreviations are used in this International Standard.

1. SPQM-RM – Software Product Quality Measurement Reference Model

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12-031
École Polytechnique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis nº 11/12-031

© ISO/IEC 2007 – All rights reserved

6 Software Product Quality Measurement

6.1 Software product quality measurement reference model (SPQM-RM)

The software product quality measurement reference model (SPQM-RM) describes the relationship between a quality model, its associated quality characteristics (and subcharacteristics), and software product attributes with the corresponding software quality measures, measurement functions, quality measure elements, and measurement methods. These relationships constitute the reference model for software product quality measurement. Figures 3 and 4 depict the relationships between the quality model and the construction of quality measures from quality measure elements. The measurement information model presented in Annex A of ISO/IEC 15939 describes the relationship between software product attributes and the measurement methods used to produce base and derived measures which, in turn, may be used as quality measure elements. The left side of Figure 3 shows that software product quality is composed of quality characteristics which in turn may be composed of subcharacteristics. The decomposition of software product quality may go through several levels although the nominal model contained in ISO/IEC 25010 only contains two levels. The right side of Figure 3 shows that software quality measures are used to indicate the quality characteristics and subcharacteristics of interest. Refer to ISO/IEC 25030 for guidance on selecting quality characteristics and subcharacteristics of interest in conjunction with the specification of quality requirements and ISO/IEC 25040 for guidance on using software quality measures for software product evaluation.

Figure 3 also shows that software quality measures are constructed by applying a measurement function to quality measure elements. In the special case where the quality measure element also serves as a software quality measure, the measurement function applied would be the identity function.

Quality measure elements may be either base or derived measures. Quality measure elements are constructed in accordance with the guidance provided in ISO/IEC 15939.

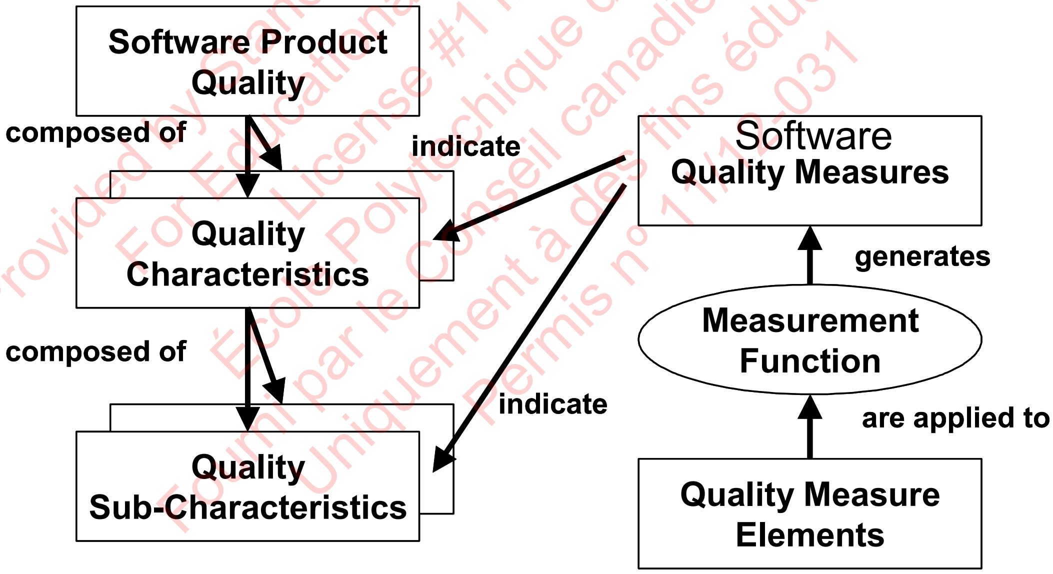

Figure 3 – Software Product Quality Measurement Reference Model (SPQM-RM)

6.2 Selecting software quality measures

Software quality measures are selected to satisfy the needs of developers, acquirers, managers, and others for information. In the context of the SQuaRE series, information needs may be defined by quality requirements specifications and product quality evaluations.

6.2.1 Criteria for selecting software quality measures and quality measure elements to fulfil those information needs shall be documented.

© ISO/IEC 2007 – All rights reserved
FINAL DRAFT / PROJET FINAL

At a minimum, the criteria shall be listed along with the assessment of the selected measures against the stated criteria. Annex A provides examples of criteria for selecting software quality measures.

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

6.2.2 When using a modified or a new measure not identified in the 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 to be constructed from quality measure elements.

Annex C provides an example of how to document a quality measure.

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

6.3 Constructing software quality measures

Three different types of software quality measures are defined to correspond to the software product quality life cycle as shown in Figure 4. Internal software quality measures are applied to a part of a software product during its development stages (such as request for proposal, requirements definition, design specification or source code). Internal software product quality measurement provides users the ability to measure the quality of the intermediate deliverables or work products. Additionally, these measurements may be used with an analysis model to predict the quality of the final software product. This allows the user to detect software product quality issues and take corrective actions during the early stages of the development life cycle process.

External software quality measures are used to measure the quality of the software product based on the behaviour of the system of which it is a part. External software product quality measurement is used only during the testing and operational stages of the product life cycle.

Measures of quality in use measure the extent to which a product meets the needs of specific users with respect to their specific personal or business goals. These measures can only be made in a realistic and operational system environment.

Throughout the product life cycle these software quality measures are made and used to assist with the successful management of product development, evaluation and support (see Figure 4).

6.3.1. The definitions of the software quality measures used shall be documented.

The definitions shall include: the quality measure name, the corresponding quality characteristic and subcharacteristic, the measurement focus (quality life cycle phase) it relates to, a purpose statement, the decision criteria for interpretation and action, and identification of the quality measure elements used to construct it. It is recommended, however, that the definition of the software quality measure contain all of the information included in the example format provided in Annex C.

NOTE An example set of internal software quality measures along with their definitions is given in ISO/IEC 25022. An example set of external software quality measures along with their definitions is given in ISO/IEC 25023. An example set of quality in use software quality measures along with their definitions is given in ISO/IEC 25024.

6.3.2. The definitions of the quality measure elements needed to construct the software quality measures shall be documented.

NOTE Quality measure elements are used throughout the software product lifecycle to construct measures of internal, external and quality in use software product quality by applying measurement methods to specified attributes and, when necessary, combining the resulting measures via a measurement function. The quality measure elements measure the attributes of the software product itself, the effects of using the software product in a specific context and the resources consumed or activities performed during software product development, testing, and maintenance. An example set of quality measure elements along with their definitions is given in ISO/IEC 25021.

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12-081
Ecole Polytechnique de Montreal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis n° 11/12-081

FINAL DRAFT / PROJET FINAL

© ISO/IEC 2007 – All rights reserved

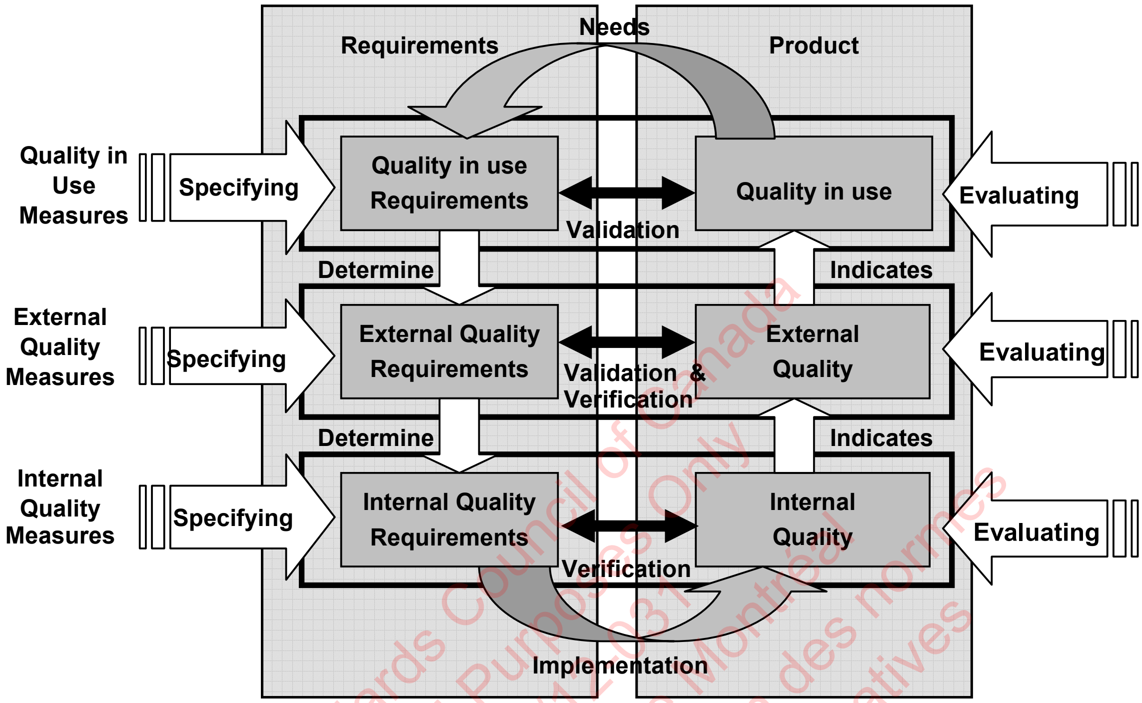

Figure 4 – Software product quality life cycle and software quality measures

NOTE Software quality measures are used to specify product requirements and to evaluate software product quality.

6.3.3 The user of the ISO/IEC 25020 international standards series shall plan and perform the measurement in order to determine values of quality measure elements and software quality measures following the reference model in Figure 3 and by using a procedure that conforms to ISO/IEC 15939.

NOTE The user of the ISO/IEC 2502n division documents for software product quality measurement is encouraged to use them in accordance with the product quality evaluation procedures contained in the ISO/IEC 2504n and the quality requirements definition process contained in the ISO/IEC 2503n divisions of standards.

Provided by Standards Council of Canada
For Educational Purposes Only
Licence #11/12-031
Ecole Polytechnique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis n° 11/12-031

Annex A
(informative)
Examples of criteria for selecting software quality measures and quality measure elements

A.1 Criteria for selecting software quality measures and quality measure elements

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

- relevance to the prioritized information needs
- repeatability and reproducibility of quality measure elements
- predictive validity of software quality measure
- feasibility of collecting the data in the organizational unit
- availability of human resources to collect and manage data
- ease of data collection
- availability of appropriate tools
- protection of privacy
- number of potentially relevant indicators supported by the required quality measure elements
- ease of interpretation by measurement users and measurement analysts
- number of users or consumers of the information products utilizing the indicator
- life cycle stage applicability
- evidence (internal or external to the organizational unit) as to the measure's fitness for purpose or information need
- characteristics of the required quality measure elements (e.g., classification dimensions in ISO/IEC 25021)

Note The user should use the results in accordance with the product quality evaluation procedures contained in the ISO/IEC 2504n and the quality requirements definition process contained in the ISO/IEC 2503n divisions of the SQuaRE standards.

Note The above list contains many criteria also recommended in ISO/IEC 15939.

A.2 Issues affecting the reliability and/or validity of measures

A.2.1 Issues affecting the measurement reliability of quality measure elements

The following issues may affect the reliability quality measure elements:

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12-031
École Polytechnique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis n° 11/12-031

FINAL DRAFT / PROJET FINAL
© ISO/IEC 2007 – All rights reserved

(a) procedures and instruments used for collecting quality measure elements

- automatically with tools or facilities/ manually collected / questionnaires or interviews

(b) source of data:

- perspective or bias of the source (e.g., developers' self reports , reviewers' reports, evaluators' reports)
- skills and abilities of those performing the data collection

A.2.2 Issues affecting the measurement validity of software quality measures

The following issue may affect the measurement validity of software quality measures:

(a) quality measure elements and associated measurement function used to produce software quality measures

- the measurement reliability of the quality measure elements used to construct the software quality measure
- quality measure elements having large correlations with measures of other software quality characteristics may confound interpretation of the desired software quality measure

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12-031
École Polytechnique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis nº 11/12-031

Annex B
(informative)
Assessing measurement validity and reliability

B.1. Assessing measurement validity

Methods for demonstrating the validity of measures typically involve both a logical argument and statistical evidence. Face validity is one type of validity. Face validity is based on logical argument or assertion that a measure is valid. Lines of Code could as a measure of size has face validity because it is logically related to common notions of size. 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 evidence 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. Some examples of systematic variation are described below.

B.1.1 Correlation

The variation in the quality characteristics values (the results of principal measures in operational use) explained by the variation in the measure values, is given by the square of the linear coefficient.

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

B.1.2 Tracking

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

NOTE An measurement user can detect movement of quality characteristics along a time period without measuring directly by using those measures which have tracking ability.

B.1.3 Consistency

If quality characteristics values (the results of principal measures in operational use) Q1, Q2,..., Qn, corresponding to products or processes 1, 2,..., n, have the relationship Q1 > Q2 > ...> Qn, then the corresponding measure values would have the relationship M1 > M2 > ...> Mn.

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

B.1.4 Predictability

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, prediction error, which is {(predicted Q(T2) - actual Q(T2)) / actual Q(T2)}, would be within the allowed prediction error range.

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

B.1.5 Discrimination

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

Provided by Standards Council of Canada
For Educational Purposes Only
License #1 1/12-031
École Polytechnique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis n° 1 1/12-031

FINAL DRAFT / PROJET FINAL
© ISO/IEC 2007 – All rights reserved

NOTE An measurement user can categorize software components and rate quality characteristics values by using those measures which have the capability to discriminate between high and low quality.

B.2. Assessing measurement reliability

Measurement reliability is most important 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 judgment as well as the conditions within which the instrumentation is applied. In the SQuaRE series, measurement reliability is primarily a concern to the selection and collection of the quality measure elements defined in ISO/IEC 25021. As noted in Annex D of ISO/IEC 15939

> [t]he reliability of a measurement method should be approached from two perspectives:
>
> - repeatability: the degree to which the repeated use of the base measure in the same Organisational 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 the repeated use of the measure in the same Organisational 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, personal differences, and so on. Various statistics have been developed to characterize measurement reliability. For measures utilizing an ordinal or nominal scale, the Kappa statistic can be used. For measures utilizing an interval or absolute scale, Cohen's alpha or other correlation-based measures can be used. More information on measurement reliability can be found within the field of measurement system evaluation.

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12-031
École Polytechnique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis n° 11/12-031

Annex C
(informative)
Example format for documenting software quality measures

The following provides an example template for documenting the software quality measures for the software product quality characteristics of importance. The ITEM column of the following table indicates the recommended content for a software product quality measure definition. The CONTENT column provides a description what should be included in this field as well a suggestions where to find content within the ISO/IEC 25000 series of standards.

ITEM	CONTENT
Software Quality Measure Name	Assigned name of the quality measure. This is taken from ISO/IEC 25022 through ISO/IEC 25024 or is provided by the user. <u>Example: Estimated latent fault density</u>
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. <u>Example: External Quality – Reliability</u>
Subcharacteristic	Quality subcharacteristic, 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. <u>Example: External Quality - Maturity</u>
Product Quality Life-cycle Phase (Measurement Focus)	Applicable portion of product quality life-cycle; internal, external or quality in use. These correspond to product quality life-cycle phases as described in ISO/IEC 25010. If the user is using a different product quality model, then the user should provide this information as applicable. <u>Example: External Quality (Testing phase)</u>
Purpose of the Software Quality Measure (Information Need)	Should be declarative statement. Often the purpose of the quality measure 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> in order to <statement of why make the measurement>. <u>Example: Evaluate the code quality by monitoring the test process and resulting fault density in order to determine the probability of satisfying the reliability requirements. Question: How many future faults might we find?</u>
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 will often be set with respect to quality requirements and corresponding evaluation criteria. Users may also 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. <u>Example: If estimated defect density exceeds the acceptable threshold, then perform additional defect detection and removal activities.</u>

FINAL DRAFT / PROJET FINAL
© ISO/IEC 2007 – All rights reserved

Indicator/Visual Display	A depiction of how the quality measurement results will be communicated to the users of the results. Typically indicators are presented in graphical or tabular format.
Measurement Function	Equation showing how the quality measure elements are combined to produce the quality measure. Example: Estimated latent defect density = (C1 – C2) / S
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 for selecting quality measure elements. Example: C1: Total number of predicted latent faults in a software product C2: Cumulative number of unique faults detected S: Product Size
Measurement Method	Describe measurement method for the quality measure elements. If these are described elsewhere such as in 25021, then 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
Data Source(s)	Describe the data source for the quality measure elements. If these are described elsewhere such as in 25021, then 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
Selection Criterion: Measurement Validity Evidence	A statement of the extent to which the quality measure meets this selection criterion and a description of the method and evidence used to make the determination. This may use an ordinal scale of high, medium, or low with respect to the relationship between the measure and the purpose. See Annex B for information regarding measurement validity. The following template can be used: "The validity of <measure> is <rating> based upon <evidence of validity> …." Example: The validity of code maturity is high based upon the logical association between fault density and code maturity: the lower the fault density, the greater the assumed maturity of the code and the more reliable it should be.
Selection Criterion: Measurement Reliability	A statement of the extent to which the quality measure meets this selection criterion and a description of the method and evidence used to make the

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12/2031
École Polytechnique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis 11/12/2031

Evidence	determination. This may use an ordinal scale of high, medium, or low based on the measurement method and underlying assumptions. There are also statistical methods for assessing 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> based upon <evidence of reliability> ….” Example: The reliability of software size is highly reliable based upon the automation employed for counting of lines of code and the adherence to coding standards.
Selection Criterion: Cost to Collect	A statement of the extent to which the quality measure meets this selection criterion and a description of the method and evidence used to make the determination. This may use an ordinal scale of high, medium, or low based on an analysis of the costs associated with collecting the quality measure elements. Examples of cost considerations include whether the data are already being collected, whether the collection will require new tools or be done 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. (It may need some additional cost, if a new prediction model is going to be developed.)
Usage Scenarios by Role	A description of how the quality measure would be used to fulfill 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: SQA 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 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 improvement in the reliability of the software as part the decision to release the code for the next phase of testing.

FINAL DRAFT / PROJET FINAL

© ISO/IEC 2007 – All rights reserved

Bibliography

ISO/IEC 25001, Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Planning and management

ISO/IEC 25021, Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality measure elements

ISO/IEC 25022, Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Measurement of internal quality

ISO/IEC 25023, Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Measurement of external quality

ISO/IEC 25024, Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Measurement of quality in use

ISO/IEC 25030, Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality requirements

ISO/IEC 25040, Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Evaluation reference model and guide

ISO/IEC 25041, Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Evaluation modules

ISO/IEC 25042, Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Evaluation process for developers

ISO/IEC 25043, Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Evaluation process for acquirers

ISO/IEC 25044, Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Evaluation process for evaluators

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12-031
École Polytechnique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis n° 11/12-031

Provided by Standards Council of Canada
For Educational Purposes Only
License #11/12-031
École Polytechique de Montréal
Fourni par le Conseil canadien des normes
Uniquement à des fins éducatives
Permis nº 11/12-031

ICS 35.080

Price based on 15 pages

© ISO/IEC 2007 – All rights reserved