

Software Engineering Processes — Software product Quality —

Part 3-1:

Requirements

DKS 2896-3-1:2019

TECHNICAL COMMITTEE REPRESENTATION

University of Nairobi
Armodio Ltd.
ICT Authority of Kenya
Daystar University
Oracle
IDEAZ Software
Intel
Kenya National Library Services
Nairobi Accelerator Lab(Nailab)
Tangazoletu
Kenya Bureau of Standards — Secretariat

REVISION OF KENYA STANDARDS

In order to keep abreast of progress in industry, Kenya Standards shall be regularly reviewed. Suggestions for improvements to published standards, addressed to the Managing Director, Kenya Bureau of Standards, are welcome.

© Kenya Bureau of Standards, 2019

Copyright. Users are reminded that by virtue of Section 25 of the Copyright Act, Cap. 12 of 2001 of the Laws of Kenya, copyright subsists in all Kenya Standards and except as provided under Section 26 of this Act, no Kenya Standard produced by Kenya Bureau of Standards may be reproduced, stored in a retrieval system in any form or transmitted by any means without prior permission in writing from the Managing Director.

Software Engineering Processes — Software Product Quality

Part 3-1

Requirements

KENYA BUREAU OF STANDARDS (KEBS)

Head Office: P.O. Box 54974, Nairobi-00200, Tel.: (+254 020) 605490, 602350, Fax: (+254 020) 604031
E-Mail: info@kebs.org, Web: <http://www.kebs.org>

Coast Region

P.O. Box 99376, Mombasa-80100
Tel.: (+254 041) 229563, 230939/40
229448 Fax: (+254 057) 21814

Lake Region

P.O. Box 2949, Kisumu-40100
Tel.: (+254 057) 23549, 22396

Rift Valley Region

P.O. Box 2138, Nakuru-20100
Tel.: (+254 051) 210553, 210555 Fax: (+254 041)

Foreword

This Standard was prepared by the KEBS Technical Committee 94 on Software Engineering, IT Service Management, IT Governance and Artificial Intelligence, under the guidance of the Standards Projects Committee, and it is in accordance with the procedures of the Kenya Bureau of Standards.

It is important to identify and specify software quality requirements as part of specifying the requirements for a software product. Software is usually part of a larger system. System requirements and software requirements are closely related and software requirements can therefore not be considered in isolation. This Kenya Standard focuses on software quality requirements, but takes a system perspective. Software quality requirements can be categorized by use of a quality model, for example the quality model defined in KS ISO/IEC 25010]. Measures of attributes of these characteristics and their subcharacteristics can be used to specify software quality requirements and evaluate the quality of a software product.

Software quality requirements address important issues of quality for software products. Software product quality requirements are needed for:

- specification (including contractual agreement and call for tender);
- planning (including feasibility analysis and translation of external software quality requirements into internal software quality requirements);
- development (including early identification of potential quality problems during development); and
- evaluation (including objective assessment and certification of software product quality).

If software quality requirements are not stated clearly, they may be viewed, interpreted, implemented and evaluated differently by different people. This may result in software which is inconsistent with user expectations and of poor quality; users, clients and developers who are unsatisfied; and time and cost overruns to rework software.

This Kenya Standard aims to improve the quality of software quality requirements. It does this by providing requirements and recommendations for quality requirements, and guidance for the processes used to define and analyse quality requirements. Application of this Kenya Standard should help ensure that software quality requirements are:

- in accordance with stakeholder needs;
- stated clearly and precisely;
- correct, complete, and consistent; and
- verifiable and measurable.

This Standard complies with the technical processes defined in KS 2896: 2019 - 1 related to quality requirements definition and analysis.

During the preparation of this standard, reference was made to the following documents:

- i) ISO IEC 25010
- ii) ISO IEC 25020
- iii) ISO IEC 25030
- iv) ISO IEC 25040
- v) ISO/IEC/IEEE 12207:2017
- vi) ISO/IEC/IEC 15288:2015

Acknowledgement is hereby made for the assistance derived from these sources.

Software Engineering processes — Software product quality — Part 3-1: Requirements

1 Scope and application

This Kenya standard provides requirements and recommendations for the specification of software product quality requirements and the selection and construction of software product quality measures. This Standard complies with the technical processes defined in KS 2896-1:2019, which are relevant for identification of stakeholder product quality needs and for analysis of software product quality requirements.

2 Conformance

Software quality requirements and measures conform to this Kenya Standard if they fulfil the requirements specified in Clause 5 and Clause 6.

3 Normative References

KS ISO/IEC 2010, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*

KS 2896 Part 2-1 :2019, *Software and Systems Engineering Processes — Requirements and Evaluation, Part 2-1 — Technical processes*

KS 2896 Part 2-2 :2019, *Software and Systems Engineering Processes — Requirements and Evaluation, Part 2-2, — Technical evaluation*

KS 2896 Part 3-2: 2019, *Software and Systems Engineering Processes — Software product Quality — Part 3-2: Product evaluation process*

4 Terms, definitions Acronyms and Abbreviations

For the purpose of this standard, the definitions given in **Annex A** of this standard shall apply. Additionally, ISO and IEC maintain terminological databases for use in standardization at the following addresses:

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

— ISO Online browsing platform: available at <http://www.iso.org/obp>

— IEEE Standards Dictionary Online: available at <http://ieeexplore.ieee.org/xpls/dictionary.jsp>

5 Software Product Quality Requirements

Process	Not Performed	Partially/ Informally performed	Planned and tracked	Well Defined	Exemption/ Justification	Reference Standards
Effective Rating	0	1	2	3		
5.1 General Requirements						
5.1.1 Software Quality Model						
i) a software quality model with a structure similar to the KS ISO/IEC 25010 quality model is used. (See also Annex B)						

DKS 2896-3-1:2019

Process	Not Performed	Partially/ Informally performed	Planned and tracked	Well Defined	Exemption/ Justification	Reference Standards
Effective Rating	0	1	2	3		
ii) The applied software quality model shall be documented						
5.2 Stakeholder requirements definition						
5.2.1 System boundaries						
i) System boundaries and constraints relevant for the system quality shall be documented						
ii) Specific concepts and terms used shall be defined or explained, in order to avoid misunderstandings of the system quality requirements.						
5.2.2 stakeholder quality requirements						
i) The purpose of the stakeholder quality requirements shall be specified.						
ii) All relevant stakeholders shall be listed.						
iii) The listed stakeholders' roles and interests shall be documented.						
iv) It shall be documented whether or not a stakeholder is taken into consideration when identifying stakeholder quality requirements						
v) . If a stakeholder is not taken into consideration the rationale should be documented.						
vi) Identified but not stated stakeholder quality requirements shall be documented.						
vii) Each stakeholder quality requirement shall be uniquely identified.						
viii) Each stakeholder quality requirement shall be uniquely identified.						

Process	Not Performed	Partially/ Informally performed	Planned and tracked	Well Defined	Exemption/ Justification	Reference Standards
Effective Rating	0	1	2	3		
ix) It should be documented that health, safety, security, environment and other stakeholder requirements and functions that relate to critical qualities are considered.						
x) Each stakeholder quality requirement shall be traceable to individual stakeholders or classes of stakeholders.						
5.2.3 Validation of stakeholder quality requirements						
i) The stakeholder quality requirements shall be validated and approved.						
ii) It shall be documented who has validated and approved the stakeholder quality requirements						
5.3 Software requirements						
5.3.1 Software boundaries						
i) The intended purpose of the software shall be documented.						
ii) The documentation should describe the role of the software in the system of which the software is an element.						
iii) The functional boundaries of the software in terms of functional behaviour and properties to be provided shall be documented						
iv) The system solution constraints on the software quality requirements shall be documented.						
v) The rationale and sources for system solution constraints should be documented as far as possible.						

DKS 2896-3-1:2019

Process	Not Performed	Partially/ Informally performed	Planned and tracked	Well Defined	Exemption/ Justification	Reference Standards
Effective Rating	0	1	2	3		
vi) Implementation constraints relevant for the software quality requirements shall be documented						
5.3.2 Software quality requirements						
i) software quality requirements shall be are documented according to the chosen quality model.						
ii) Software quality requirements shall be uniquely identified.						
iii) Software quality requirements shall be traceable to stakeholder requirements.						
iv) Software quality requirements shall take all relevant stakeholder quality requirements into consideration.						
v) Software quality requirement shall be associated with quality characteristics (or subcharacteristics) as defined in the applied quality model.						
vi) If no software quality requirements address a specific characteristic (or subcharacteristic) of the quality model, this shall be documented						
vii) Software quality requirements should be categorised according to the quality model as one of the following:						
○ Software quality in use requirement;						
○ External software quality requirement;						
○ Internal software quality requirement.						

Process	Not Performed	Partially/ Informally performed	Planned and tracked	Well Defined	Exemption/ Justification	Reference Standards
Effective Rating	0	1	2	3		
viii) A software quality requirement shall be specified in terms of a software quality measure and associated target value.						
ix) The software quality measures used shall be documented according to requirements in ISO/IEC 25020.						
x) It shall be documented which required functions of the software a software quality requirement is applicable for.						
xi) The criteria applied for selecting software quality measures shall be documented						
xii) The operational profile for a software quality requirement shall be specified when relevant.						
xiii) When specifying the target value of a software quality requirement the acceptable tolerance shall be documented.						
5.4 Verification of software quality requirements						
i) The software quality requirements shall be verifiable.						
ii) When special tools, techniques, or other resources such as effort or time are needed for verification this shall be documented. The time and effort needed to verify a software quality requirement should be documented. If special resources or competences are required in order to verify a software quality requirement, such demands should be documented.						

DKS 2896-3-1:2019

Process	Not Performed	Partially/ Informally performed	Planned and tracked	Well Defined	Exemption/ Justification	Reference Standards
Effective Rating	0	1	2	3		
iii) Identified conflicts between software quality requirements shall be documented						
iv) Identified conflicts between software quality requirements and implementation constraints shall be documented.						
v) Additional or changed software quality requirements that solve identified issues with software quality requirements shall be traceable to original software quality requirements. Software quality requirements that have been replaces or deleted shall be marked as such.						
vi) The software quality requirements shall be reviewed and approved.						
vii) It shall be documented who has reviewed and approved the software quality requirements.						
viii) The software quality requirements shall be documented in a format such that they can be managed according to a configuration and change management system.						

5.5 Software requirements

The requirements and recommendations in this clause assumes that high level architectural decisions have been made,

Note KS ISO/IEC 15288 describes the requirements analysis process activities.

6 Software Product Quality Measurement

Process	Not Performed	Partially/ Informally performed	Planned and tracked	Well Defined	Exemption Justification	Reference Standards
Effective Rating	0	1	2	3		
6.1 Software product quality measurement reference model						
i) The software product quality measurement reference model (SPQM-RM) shall 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.						
6.2 Selecting software quality measures						
i) At a minimum, the criteria shall be listed along with the assessment of the selected measures against the stated criteria.						
ii) When using a modified or a new measure not identified in the KS ISO/IEC 25022, KS ISO/IEC 25023 or KS 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.						
6.3 Constructing software quality measures						
i) The definitions of the software quality measures used shall be documented.						
ii) The definitions of the quality measure elements needed to						

DKS 2896-3-1:2019

Process	Not Performed	Partially/ Informally performed	Planned and tracked	Well Defined	Exemption Justification	Reference Standards
Effective Rating	0	1	2	3		
construct the software quality measures shall be documented.						
iii) The user of this Kenya Standard shall plan and perform the measurement in order to determine values of quality measure elements and software quality measures following the reference model in Annex B and by using a procedure that conforms to KS ISO/IEC 15939.						

Annex A (normative)

Terms and definition

A.1

acquirer

individual or organisation that acquires or procures a system, software product or software service from a supplier

Note Based on the definition in ISO/IEC 12207:1995.

A.2

analysis model

algorithm or calculation combining one or more base and/or derived measures with associated decision criteria

A.3

attribute

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

Note 1 based on ISO/IEC 15939:2002.

Note 2 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.

A.4

attribute for quality measure

Attribute that relates to software product itself, to the use of the software product or to its development process

Note Attributes for quality measure are used in order to obtain measurement primitives.

A.5

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].

A.6

commercial-off-the-shelf software product

software product defined by a market-driven need, commercially available, and whose fitness for use has been demonstrated by a broad spectrum of commercial users

A.7

context of use

DKS 2896-3-1:2019

users, tasks, equipment (hardware, software and materials), and the physical and social environments in which a product is used

[ISO 9241-11:1998]

A.8

custom software

software product developed for a specific application from a user requirements specification

A.9

data

collection of values assigned to base measures, derived measures and/or indicators

[ISO/IEC 15939:2002]

A.10

decision criteria

thresholds, targets, or patterns used to determine the need for action or further investigation, or to describe the level of confidence in a given result.

[ISO/IEC 15939:2002]

A.11

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.

A.12

developer

individual or organisation that performs development activities (including requirements analysis, design, testing through acceptance) during the software life cycle process

Note Based on the definition in ISO/IEC 12207:1995

A.13

division of standards

division forms a family of standards serving complementary purposes

A.14

end user

individual person who ultimately benefits from the outcomes of the system

Note The end user may be a regular operator of the software product or a casual user such as a member of the public.

A.15

entity

object that is to be characterised by measuring its attributes

EXAMPLE An object can be a process, product, project, or resource.

[ISO/IEC 15939:2002]

A.16**evaluation method**

procedure describing actions to be performed by the evaluator in order to obtain results for the specified measurement applied to the specified product components or on the product as a whole

A.17**evaluation module**

package of evaluation technology for measuring software quality characteristics, subcharacteristics or attributes

Note The package includes evaluation methods and techniques, inputs to be evaluated, data to be measured and collected and supporting procedures and tools.

A.18**evaluator**

individual or organisation that performs an evaluation

A.19**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.

A.20**failure**

termination of the ability of a product to perform a required function or its inability to perform within previously specified limits

Note Based on the definition in IEEE 610.12-1990.

A.21**fault**

incorrect step, process or data definition in a computer program

[IEEE 610.12-1990]

A.22**functional requirement**

requirement that specifies a function that a system or system component must be able to perform

[IEEE 610.12-1990]

Note The quality characteristic functionality can be used to specify or evaluate the suitability, accuracy, interoperability, security and compliance of a function (see ISO/IEC 9126-1 [ISO/IEC 25010]).

A.23**implied needs**

needs that may not have been stated but are actual needs

Note Some implied needs only become evident when the software product is used in particular conditions.

EXAMPLE Implied needs include: needs not stated but implied by other stated needs and needs not stated because they are considered to be evident or obvious.

A.24

DKS 2896-3-1:2019

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".

A.25

information need

insight necessary to manage objectives, goals, risks, and problems

[ISO/IEC 15939:2002]

A.26

information product

one or more indicators and their associated interpretations that address information need

EXAMPLE A comparison of a measured defect rate to planned defect rate along with an assessment of whether or not the difference indicates a problem.

[ISO/IEC 15939:2002]

A.27

information system needs

needs that can be specified as quality requirements by external measures and sometimes by internal measures

A.28

intermediate software product

product of the software development process that is used as input to another stage of the software development process

EXAMPLE Intermediate software products can include static and dynamic models, other documents and source code.

A.29

intermediate software product needs

needs that can be specified as quality requirements by internal measures

A.30

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.

A.31

maintainer

individual or organisation that performs maintenance activities

Note Based on the definition in ISO/IEC 12207: 1995

A.32

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]

A.33**measure (verb)**

make a measurement

[ISO/IEC 14598-1:1999]

A.34**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.).

A.35**measurement function**

algorithm or calculation performed to combine two or more base measures

[ISO/IEC 15939:2002]

A.36**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].

A.37**measurement procedure**

set of operations, described specifically, used in the performance of a particular measurement according to a given method

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

A.38**measurement process**

process for establishing, planning, performing and evaluating software measurement within an overall project or organisational measurement structure

[ISO/IEC 15939:2002]

A.39**observation**

instance of applying a measurement procedure to produce a value for a base measure

[ISO/IEC 15939:2002]

A.40**operator**

individual or organisation that operates the system

DKS 2896-3-1:2019

Note Based on the definition in ISO/IEC 12207:1995.

A.41

process

system of activities, which use resources to transform inputs into outputs

[ISO 9000:2000]

A.42

quality in use (measure)

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

A.43

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.

A.44

quality model

defined set of characteristics, and of relationships between them, which provides a framework for specifying quality requirements and evaluating quality

A.45

rating

action of mapping the measured value to the appropriate rating level. Used to determine the rating

level associated with the software product for a specific quality characteristic

A.46

rating level

scale point on an ordinal scale, which is used to categorise a measurement scale

Note 1 The rating level enables software product to be classified (rated) in accordance with the stated or implied needs.

Note 2 Appropriate rating levels may be associated with the different views of quality i.e. Users', Managers' or Developers'.

A.47

requirements

expression of a perceived need that something be accomplished or realized

Note The requirements may be specified as part of a contract, or specified by the development organisation, as when a product is developed for unspecified users, such as consumer software, or the requirements may be more general, as when a user evaluates products for comparison and selection purpose.

A.48

scale

ordered set of values, continuous or discrete, or a set of categories to which the attribute is

mapped

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

EXAMPLE Types of scales are: a nominal scale which corresponds to a set of categories; an ordinal scale which corresponds to an ordered set of scale points; an interval scale which corresponds to an ordered scale with equidistant scale points; and a ratio scale which not only has equidistant scale point but also possesses an absolute zero. Measures using nominal or ordinal scales produce qualitative data, and measures using interval and ratio scales produce quantitative data.

A.49**software product**

set of computer programs, procedures, and possibly associated documentation and data

[ISO/IEC 12207:1995]

Note 1 Products include intermediate products, and products intended for users such as developers and maintainers.

Note 2 In SQuaRE International Standards software quality has the same meaning as software product quality.

A.50**software product evaluation**

technical operation that consists of producing an assessment of one or more characteristics of a software product according to a specified procedure

A.51**software quality**

capability of software product to satisfy stated and implied needs when used under specified conditions

Note This definition differs from the ISO 9000:2000 quality definition mainly because the software quality definition refers to the satisfaction of stated and implied needs, while the ISO 9000 quality definition refers to the satisfaction of requirements.

A.52**software quality characteristic**

category of software quality attributes that bears on software quality

Note Software quality characteristics may be refined into multiple levels of subcharacteristics and finally into software quality attributes.

A.53**software quality evaluation**

systematic examination of the extent to which a software product is capable of satisfying stated and implied needs

A.54**software quality in use**

capability of the software product to enable specific users to achieve specific goals with effectiveness, productivity, safety and satisfaction in specific contexts of use

Note Before the product is released, quality in use can be specified and measured in a test environment for the intended users, goals and contexts of use. Once in use, it can be measured for actual users, goals and contexts of use. The actual needs of users may not be the same as those anticipated in requirements, so actual quality in use may be different from quality in use measured earlier in a test environment.

A.55**Software quality measure**

measure of internal software quality, external software quality or software quality in use

Note Internal software quality, external software quality and software quality in use are described in the quality model in ISO/IEC 9126-1 [ISO/IEC 25010].

A.56**stakeholder**

DKS 2896-3-1:2019

a party having a right, share or claim in a system or in its possession of characteristics that meet that party's needs and expectations

[ISO/IEC 15288:2002]

Note Stakeholders include, but are not limited to, end users, end user organisations, supporters, developers, producers, trainers, maintainers, disposers, acquirers, supplier organisations and regulatory bodies

A.57

supplier

individual or organisation that enters into a contract with the acquirer for the supply of a system, software product or software service under the terms of the contract

[ISO/IEC 12207:1995]

A.58

system

a combination of interacting elements organised to achieve one or more stated purposes

Note 1 A system may be considered as a product or as the services it provides.

Note 2 In practice, the interpretation of its meaning is frequently clarified by the use of an associative noun, e.g. aircraft system. Alternatively, the word system may be substituted simply by a context dependent synonym, e.g. aircraft, though this may then obscure a system principles perspective.

[ISO/IEC 15288:2002]

A.59

target of process

software product or task executed by software product to which measurement or evaluation process is applied

A.60

unit of measurement

particular quantity defined and adopted by convention, with which other quantities of the same kind are compared in order to express their magnitude relative to that quantity

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

A.61

user

individual or organisation that uses the system to perform a specific function

Note Users may include operators, recipients of the results of the software, or developers or maintainers of software.

[ISO/IEC 15939:2002]

A.62

validation

confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled

Note 1 "Validated" is used to designate the corresponding status.

[ISO 9000:2000]

Note 2 In design and development, validation concerns the process of examining a product to determine conformity with user needs.

Note 3 Validation is normally performed on the final product under defined operating conditions. It may be necessary in earlier stages.

Note 4 Multiple validations may be carried out if there are different intended uses.

A.63

value

number or category assigned to an attribute of an entity by making a measurement

A.64

verification

confirmation, through the provision of objective evidence, that specified requirements have been fulfilled

Note 1 "Verified" is used to designate the corresponding status.

[ISO 9000:2000]

Note 2 In design and development, verification concerns the process of examining the result of a given activity to determine conformity with the stated requirement for that activity.

PUBLIC REVIEW DRAFT OCTOBER 2019

Annex B

(normative)

Software Products Quality Models

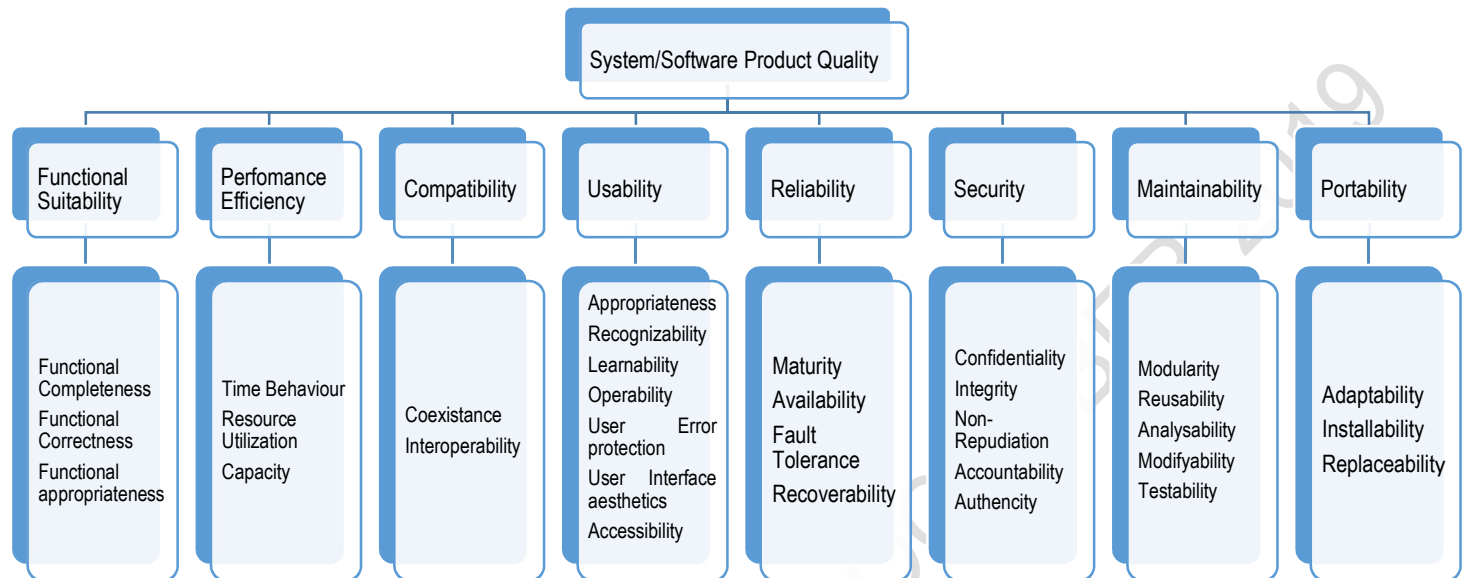


Figure B.1 — Product quality model

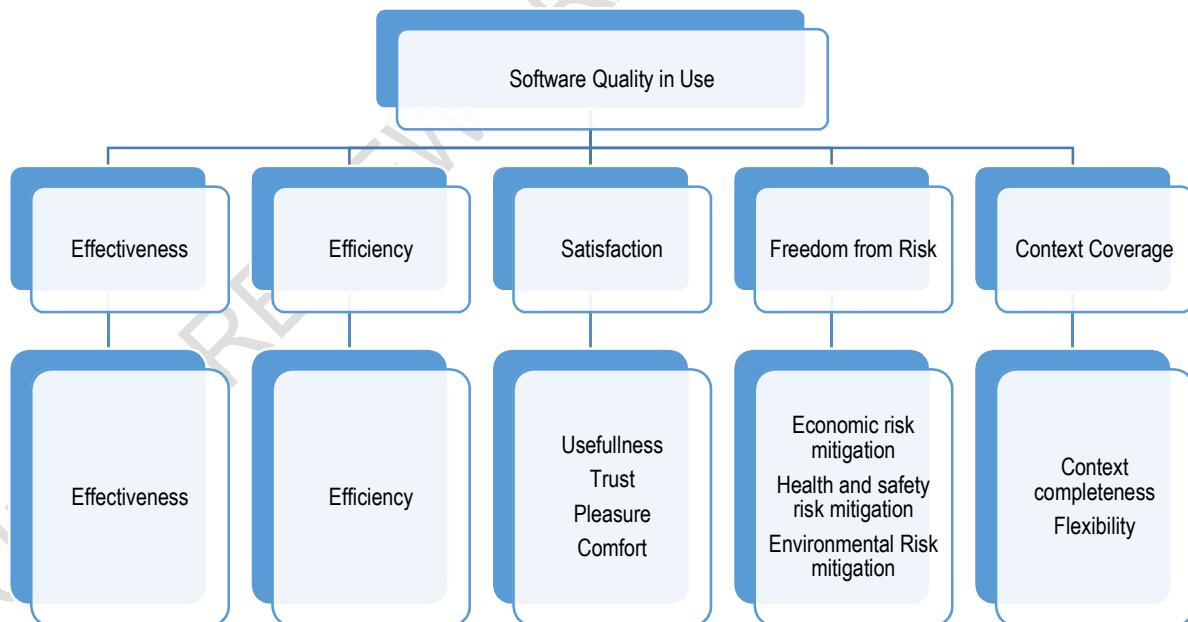


Figure B.2 — Quality in use model

Annex C
(normative)

Template 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 as suggestions where to find content within the ISO/IEC 25000 series of standards.

ITEM	CONTENT	REFERENCES
Software Quality Measure Name	Assigned name of the quality measure.	KS ISO/IEC 25022 through KS ISO/IEC 25024 or is provided by the user.
Software Product Quality Characteristic	Quality characteristic from the quality model used.	KS ISO/IEC 25022 through ISO/IEC 25024 or is provided by the user based on the quality model being used.
Subcharacteristic	Quality subcharacteristic, if applicable	ISO/IEC 25022 through ISO/IEC 25024 or is provided by the user based on the quality model being used.
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. Example: External Quality (Testing phase)	
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>. 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?	
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. Example: If estimated defect density exceeds the acceptable	

DKS 2896-3-1:2019

ITEM	CONTENT	REFERENCES
	threshold, then perform additional defect detection and removal activities.	
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, 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	KS ISO/ IEC 25021
Data Source(s)	Describe the data source(s) for the quality measure elements. If these are described elsewhere such as in ISO/IEC 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	<p>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</p> <p>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.</p> <p>This may use an ordinal scale of high, medium, or low with respect to the relationship between the measure and the purpose. The following template can be used: The validity of <measure> is <rating> based upon <evidence of validity> ...</p> <p>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.</p>	

ITEM	CONTENT	REFERENCES
<p>Selection Criterion: Measurement Reliability Evidence</p>	<p>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.</p> <p>The following template can be used: The reliability of <measure> is <rating> based upon <evidence of reliability> ...</p> <p>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.</p>	
<p>Selection Criterion: Cost to Collect</p>	<p>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.</p> <p>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.)</p>	