INTERNATIONAL STANDARD

ISO/IEC 14598-3

First edition
2000-02-01

Software engineering — Product evaluation —

Part 3: Process for developers

Ingénierie du logiciel — Évaluation du produit —

Partie 3: Procédés pour développeurs

Reference number
ISO/IEC 14598-3:2000(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 734 10 79
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents

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. 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.

International Standard ISO/IEC 14598-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software engineering*.

ISO/IEC 14598 consists of the following parts, under the general title *Software engineering — Product evaluation*:

— *Part 1: General overview*

— *Part 2: Planning and management*

— *Part 3: Process for developers*

— *Part 4: Process for acquirers*

— *Part 5: Process for evaluators*

— *Part 6: Documentation of evaluation modules*

Annex A of this part of ISO/IEC 14598 is for information only.

Introduction

This part of ISO/IEC 14598 is intended for use during software development. It is applicable to all software development activities requiring a disciplined process. This part of ISO/IEC 14598 is particularly aimed at those measuring and evaluating the quality of software.

This part of ISO/IEC 14598 provides guidelines for clarifying quality requirements and for implementing and analysing software quality measures. This part of ISO/IEC 14598 applies to all software at all phases of the development life cycle. It focuses on the selection and reporting of those indicators that are useful to predict end product quality by measuring the quality of intermediate products. It also focuses on measuring end product quality.

Software engineering - Product evaluation - Part 3: Process for developers

1 Scope

This part of ISO/IEC 14598 provides requirements and recommendations for the practical implementation of software product evaluation when the evaluation is conducted in parallel with the development and carried out by the developer. In particular, it may be used to apply the concepts described in ISO/IEC 9126-1, 2, 3 and ISO/IEC 14598-1, 2, 6.

The process described in this part of ISO/IEC 14598 defines the activities needed to analyse evaluation requirements, to specify, design, and perform evaluation actions and to conclude the evaluation of any kind of software product.

The evaluation process is designed to be used concurrently with the development. The evaluation process needs to be synchronised with the software development process and the entities be evaluated as they are delivered.

This part of ISO/IEC 14598 may be used by

- a project manager to clarify quality requirements, to monitor and control the quality of the software during development and to make decisions to assure that the required quality is built in,
- a software designer to identify specific features that should be built into the software or changed in order to meet the quality requirements,
- a quality assurance / control / audit responsible to evaluate whether the quality requirements are met,
- a maintainer to make decisions for the implementation of changes and redesign/reengineering,
- a software acquirer as part of an agreement with a developer when acquiring software (e.g. in the case of outsourcing software development) when an independent evaluation is not required. Acquirers may be personnel in a purchasing role, developers outsourcing a part of the software product, or end-users. The role of the acquirer depends on the agreement between the acquirer and the developer. ISO/IEC 14598-4 describes evaluation from the acquirers point of view.

This part of ISO/IEC 14598 is intended for application at the project level. In order to obtain full benefit from this standard the organization should be involved. This aspect is covered in ISO/IEC 14598-2.

This part of ISO/IEC 14598 does not prescribe specific indicators or metrics nor does it prescribe any particular development method.

2 Conformance

In order to conform to this part of ISO/IEC 14598, an organization shall review all requirements and recommendations in clause 6, to identify which are applicable, and state which requirements have not been implemented.

3 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 14598. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 14598 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 9126-1, *Information technology - Software product quality - Part 1: Quality model.*

ISO/IEC 12207, *Information technology - Software life cycle processes.*

ISO/IEC 14598-1:1999, *Information technology - Software product evaluation - Part 1: General overview.*

ISO/IEC 14598-2:2000, *Information technology - Software product evaluation - Part 2: Planning and management.*

ISO/IEC 14598-6, *Software engineering - Product evaluation - Part 6: Documentation of evaluation modules.*

4 Terms and definitions

For the purposes of this part of ISO/IEC 14598, the definitions given in ISO/IEC 14598-1 and the following definitions apply.

4.1
counting rule
conditions and procedures under which the measurement value is obtained

4.2
external attribute
a measurable property of an entity which can only be derived with respect to how it relates to its environment

Note: External attributes are those that relate to requirements (external properties of the software). External attributes can only be derived from the operational behaviour of the system of which it is a part.

4.3
internal attribute
a measurable property of an entity which can be derived purely in terms of the entity itself

Note: Internal attributes are those that relate to the internal organization of the software and its development.

4.4
unit
a quantity adopted as a standard of measurement

Note: Each unit has an associated scale.

5 Evaluation concepts

5.1 General aspects

The quality of software products can be described in terms of quality characteristics.

Note: A set of quality characteristics is defined in ISO/IEC 9126-1.

However, in general it is not practical to assign measurement values directly to these characteristics. Instead, a set of software quality attributes of the software product is selected that represents the main aspects of the characteristics. Measurement values of these attributes give a quantitative representation of the quality of the software product.

The emphasis of this part of ISO/IEC 14598 is to support the developer when applying software measurement and evaluation during the development life cycle. This is done by identifying attributes of the intermediate products and development activities and by measuring these attributes. This provides a means for quantitatively monitor and control the quality of the software product under development during the development process. The goal is to identify problems in achieving the desired quality as early as possible in the development process.

Today's knowledge of software measurement and evaluation does not justify the recommendation of a single set of attributes that apply to every software product and every software developing organization. Therefore, the selection of attributes of the software product, the intermediate products and development activities is based on the experience of the organization developing the software.

5.2 User needs

Identification of the user needs is an important aspect of establishing general quality requirements. This is done by identifying the user needs for quality in use in particular contexts of use. These general requirements are informal by nature and needs to be formalised. They can be quantified and evaluated using quality in use metrics.

Note : A set of quality in use metrics is described in ISO/IEC 9126-4.

The approach taken in this part of ISO/IEC 14598 is to formulate the general requirements in terms of external attributes.

5.3 External attributes

External quality attributes represent the quality characteristics of the software product. They are used to express external quality requirements quantitatively. This is done by assigning a target measurement value to each attribute.

When the software product is developed actual measurement values of the attributes are collected, hereby providing a quantitative expression of the quality characteristics of the software. Quality evaluation is done by comparing the actual measured values with the target values of all attributes.

Note: A set of external software quality metrics is provided in ISO/IEC 9126-2.

5.4 Internal attributes

In order to monitor and control the software quality during development the external quality requirements are translated into requirements of intermediate products and development activities. This is done by translating the target measurement values of the external attributes of the software product into target measurement values of internal attributes of intermediate products and development activities.

The selection of internal attributes and translation of external target values into internal target values is a non-trivial activity. It depends primarily on personal experience unless the developer provides an infrastructure for collecting and analysing experience from previously completed projects. In that case, the experience of the developer can support the activity.

Note 1:The organizational aspect is described in ISO/IEC 14598-2.

During development actual values of internal attributes are measured. The values are compared to the target values. This provides a control of the software quality during development.

Internal attributes can be used to identify anomalies or outliers (i.e. attribute values that deviate from what would normally be expected). General experience tells that such entities are worth examining more closely.

Some internal attributes can be used to monitor trends in the development when they are measured periodically (e.g. every week). Trends measures are used for identifying problems early, both related to the product and to the development process.

Note 2: A set of internal metrics is provided in ISO/IEC 9126-3.

5.5 Quality indicators

Internal quality attributes can be used as quality indicators. In particular, internal attributes are often used as indicators of external attributes; but no general, direct relationship between quality indicators and external quality attributes has been validated yet. However, it is commonly accepted that quality indicators provide useful guidance when used with care.

Use of quality indicators allows the software developer to identify possible quality problems early in the development and to take corrective actions immediately.

There is no known universal set of quality indicators that is suitable for every software development effort. There are differences in applications, development methods and tools, project organizations and cultural differences to mention some examples. Therefore, some indicators may be useful in one organization, but not work in another organization.

5.6 Evaluation process

The evaluation process described in this part of 14598 consists of a set of activities which are conducted by the developer. These activities are performed on the basis of measurement values obtained during the development process.

Note 1: The generic evaluation process is described in ISO/IEC 14598-1.

Note 2 : The organizational aspects of the evaluation is described in ISO/IEC 14598-2.

The evaluation process comprises the five activities listed below:

- Establishment of evaluation requirements which consists of identifying the general quality requirements according to an agreed quality model. This activity is described in 6.2.

- Specification of the evaluation which consists of determining the external metrics and target measurement values (criteria for evaluation). This activity is described in 6.3.1. Specification also consists of determining the internal metrics and target measurement values (criteria for evaluation). This activity is described in 6.3.2.

- Design of the evaluation which consists of planning data collection actions. This activity is described in 6.4.1 and 6.4.2.

- Execution of the evaluation which consists of collecting internal measurement values during development and comparing them with target values (evaluation during development). Internal attribute values (quality indicators) are used to estimate end product quality. This is described in 6.5.1. It also consists of collecting external measurement values when they become available and comparing with target values (evaluation of product quality). This activity is described in 6.5.2.

- Feedback to the organization which is based on a of review of evaluation results. This activity is described in 6.6.

5.7 Relation between evaluation and life cycle processes

Evaluation of a software product can be performed within the context of any life cycle process.

Note 1: Software life cycle processes are defined in ISO/IEC 12207: 1995.

This part of ISO/IEC 14598 relates primarily to the development processes.

Note 2 : Development processes are described in ISO/IEC 12207 subclause 5.3. As stated in ISO/IEC 12207, this implies that it may also be necessary to consider the maintenance process (subclause 5.5) and supporting life cycle processes (clause 6) and organizational life cycle processes (clause 7). When this standard is used in the case of outsourcing software development it also relates to the acquisition process and the supply process as described in ISO/IEC 12207 subclause 5.1 and 5.2

6 Evaluation process requirements

6.1 General requirements

This clause relates to the organizational and project-specific requirement.

6.1.1 Organizational requirements

The developer shall build up an infrastructure that allows for data collection and process modifications based on data analysis.

Note : Organizatorial aspects of evaluation are described in ISO/IEC 14598-2

6.1.2 Project requirements

The developer shall develop the software following a disciplined development process that allows for planning and conducting software measurement and evaluation.

Note 1: Life cycle processes are described in ISO/IEC 12207. Development are described in subclause 5.3.

Note 2: An overview of software product evaluation can be found in ISO/IEC 14598-1.

The developer shall coordinate evaluation activities with supporting processes and activities.

Note 3 : Supporting processes are described in ISO/IEC 12207, including in particular the quality assurance process (subclause 6.3), the verification process (subclause 6.4), the validation process (subclause 6.5) and the audit process (subclause 6.7).

Many data analysis methods require data from previous projects developed under similar conditions and with comparable quality requirements. The developer should, therefore, apply a development model similar to one that has been used in previous projects in the developers organization. Also the same set of attributes should be applied in the projects to allow for data analysis.

6.2 Establish evaluation requirements

This clause relates to the establishment of the general quality requirements and the analysis of their feasibility.

6.2.1 Quality requirements identification

The developer shall ensure that general quality requirements applicable to the software system are identified. User needs, organizational experience, application area experience, software integrity requirements, required standards, regulations, laws etc. should be considered when identifying general requirements.

Note 1: Software integrity levels are described in ISO/IEC 15026.

The developer shall ensure that an agreed quality model is used for structuring the quality requirements.

Note 2 : A quality model is described in ISO/IEC 9126-1.

A list of other system requirements that may affect the feasibility of the quality requirements shall be produced. Acquisition concerns, such as cost and schedule constraints, warranties, and organizational concerns should be considered. Mutually exclusive requirements should be resolved.

Note 3: Focus should be on external product attributes.

All parties involved in the creation and use of the software system should participate or be represented in the quality requirements identification process.

The relative priorities of the requirements should be discussed with all involved parties. Each group should weigh the quality requirements against the other system requirements and constraints. All viewpoints should be considered.

The identified quality requirements may be conflicting or cooperative. Conflicts between the requirements shall be resolved. In addition, if the choice of quality requirements is in conflict with cost, schedule or system functionality, one or the other shall be altered.

The developer shall execute a feasibility analysis of the quality requirements. Experience from previous projects with similar quality requirements carried out in the developers organization should be considered.

The developer shall ensure that the requirements are technically feasible, reasonable, complementary, achievable and verifiable.

The quality requirements shall be resolved into a single set of quality requirements formulated according to the agreed quality model. Agreement to the final list of general requirements should be sought from all involved parties.

6.3 Specification of the evaluation

This subclause relates to quantification of quality requirements. For each requirement, one or more external attributes are selected to represent the requirement. Assigned target values serve as quantitative representations of the requirements (evaluation criteria).

For each external requirement, one or more internal attributes are selected to represent that requirement during the development. Assigned target values to the internal attributes are used to control the quality during the development.

6.3.1 External quality requirements

The developer shall define in which life cycle processes and activities the measurement and evaluation will be implemented.

Note 1: The measurement and evaluation of external attributes will normally take place after development has been completed.

The developer shall define which entities are to be measured and evaluated.

Note 2: The entities will normally be part of the end product (e.g. the running system or user manual).

The developer shall define which external attributes are to be measured.

The developer shall identify metrics for each quality requirement (from the defined external attributes and entities).

The developer shall define target values for each metric.

Note 3: The target values give a quantitative representation of the quality requirements.

Note 4: The target values are used as the evaluation criteria.

The developer shall define conditions under which the measurement is to be performed. This means identifying other attributes whose values influence the measurement and defining the values of these attributes.

The developer shall execute a refined feasibility analysis of the quality requirements. Experience from previous projects with similar quality requirements carried out in the developers organization should be considered.

The developer shall ensure that the requirements are technically feasible, reasonable, complementary, achievable and verifiable.

External attribute values may depend on the value of other attributes. These conditions shall be specified to make measurement values meaningful.

Note 5: For example, the response time of a system depends on the hardware, operating system, other programs running on the system, user profile, etc.

6.3.2 Internal quality requirements

The developer shall define in which life cycle processes and activities the measurement and evaluation of the internal attributes will be implemented.

Note 1: The measurement and evaluation of the internal attributes will normally take place during the development process.

The developer shall define which entities are to be measured and evaluated.

Note 2: The selected entities will normally be intermediate products and activities.

The developer shall define which internal attributes are to be measured.

Note 3: For different intermediate products different attributes may be needed.

The developer shall identify metrics for each relevant combination of attributes and entities.

The developer shall define a set of internal attributes that

- covers every relevant intermediate product and activity,

- are appropriate for the application domain and for the method to be used in the development,

- cover identified product and development risks.

Note 4: Examples of development risks include unstable specifications, identified problems not being resolved, running behind schedule, etc.

Appropriate trend measures should be included.

Note 5: When they are applied periodically, some metrics are useful for identifying trends in the software development process. Examples of such trend measures are 'Number of completed modules', 'Number of resolved problems', 'Number of changed requirements', etc.

The developer shall define a set of internal attributes that relate to all external attributes ; i.e., to all quality requirements. These attributes are used as quality indicators.

Note 6: Relevant intermediate products should be analyzed and internal measurement data be collected for two purposes:

- evaluating the quality of the intermediate products to find indications of the fulfilment (or non fulfilment) of their quality requirements.

- getting an indication (prediction) of the quality of an end product.

Note 7: ISO/IEC 9126-3 can be used as guidance for selecting indicators.

The developer shall describe the predictive model for the defined quality indicator ; i.e., the relationship between the indicators and the external quality attributes.

Note 8: An indicator does not require a rigid one to one relationship with the quality attribute it seeks to measure. However, the link between the indicator(s) and the relevant quality attribute(s) should be clearly defined.

For efficient management use, the number of indicators should be kept low. Priority should be given to indicators that can be supported by data already collected during existing processes, such as configuration management or integration testing.

The developer shall set target values for the internal attributes when appropriate.

The developer shall define conditions under which the measurement is to be performed. This means identifying other attributes whose value influence the measurement and define the values of these attributes.

Note 9: By definition the value of an internal attribute can be measured independently of other attributes.

6.4 Design of the evaluation

This clause relates to design of the evaluation. The external evaluation concerns the external quality requirements, and the internal evaluation concerns the internal quality monitoring and control during development.

Note : A reference to a quantitative evaluation plan may be found in ISO/IEC 14598-2.

6.4.1 Planning the external evaluation

The developer shall specify data collection actions (procedures) to be performed for obtaining actual values for each external metric. This includes specification of time schedules, responsibilities, and use of data collection and analysis tools. If special training of personnel is required, this should also be planned.

The developer shall define the measurement precision. Any statistical models applied shall be specified, including input data requirements, sampling strategies etc.

Note: If the developer's organization has defined a set of evaluation modules this activity also includes selecting evaluation modules. Documentation of evaluation modules is described in ISO/IEC 14598-6.

6.4.2 Planning the internal evaluation

The developer shall specify data collection actions (procedures) to be performed to obtain actual values for each internal metric. This includes specification of time schedules, responsibilities, and use of data collection and analysis tools. If special training of personnel is required, this should also be planned.

The developer shall define the measurement precision. Any statistical models applied shall be specified, including data requirements, sampling strategies etc.

The developer shall define contingency actions, like extra evaluation, if measurement results are inconclusive or alarming.

The developer shall consider any influences on the software development activities. The set of measurements may imply a change in the development process, through its need for data acquisition.

Note 1: Hardware or software tools may have to be located, evaluated, purchased, adapted or developed to implement the measurements. The set of measurements may imply a change in the organizational structure used to produce the software system. The quality assurance / control organization or the entire development team may need training in the use of the measurements and data collection procedures. If the implementation of measurements has caused changes in the development process, the development team may need to be educated about the changes.

Note 2: If the developers organization has defined a set of evaluation modules, this activity also includes selecting evaluation modules. Documentation of evaluation modules is described in ISO/IEC 14598-6.

6.5 Execution of the evaluation

This clause relates to collecting the quality data as planned and to comparing with target values (evaluation criteria).

6.5.1 Internal evaluation

Quality monitoring and control takes place during the development. Actual values for the internal attributes are collected. In case of undesirable values, the cause is analysed, thereby allowing the developer to understand and react to problems.

The developer shall collect actual measurement values for defined internal attributes according to the defined data collection actions. If the quality requirements are changed, the developer shall reconsider the specification of the evaluation (6.3) and the design of the evaluation (6.4).

The developer shall take necessary actions to ensure the quality of the collected data. The actions should, when appropriate, include validating automated tools for data collection and checking data by human procedures.

The developer shall compare actual values with target values when they are assigned.

The developer should use actual values of defined indicators to estimate final product quality. Experience from the development organization's previous projects with similar quality requirements should be taken into account.

Note: Quality prediction is dependent on validated indicators. A development organization will first need to collect indicator values and product measurement values for several projects to get a set of validated indicators.

The developer should use actual values to monitor trends in order to identify development risks.

The developer should analyse actual values in order to identify outlier values. Outlier values often indicate problems or unusual conditions. Explanation of outlier values should always be sought. Sometimes there are good reasons for outlier values. In that case, there may be no reason for corrective actions.

Contingency actions shall be taken when necessary.

6.5.2 Evaluation of the end product

Quality evaluation of the software product takes place when the development has been completed. Actual values for the external attributes are collected.

Note 1: If possible, components of the software may be measured before the development is completed.

The developer shall collect actual measurement values for defined external attributes according to the defined data collection actions. If the quality requirements are changed, the developer shall reconsider the specification of the evaluation (6.3) and the design of the evaluation (6.4).

The developer shall take necessary actions to ensure the quality of the collected data. The actions should, when appropriate, include validating automated tools for data collection and checking data by human procedures.

The developer shall compare actual values with target values (evaluation criteria).

Note 2: The evaluation process described in this international standard is conducted by the developer. ISO/IEC 14598-5 describes an independent evaluation process.

The developer shall make an assessment of the evaluation results. The actual values should be summarised and compared with other values such as time and cost in order to support a decision on the result of the development (e.g. improving the product, reviewing the requirements, etc).

The developer should document the evaluation result.

6.6 Quality evaluation review and feedback to the organization

The developer shall make the data collected available to the organization for use in other development projects.

The developer shall review the results of the evaluation and the validity of the evaluation process, indicators and metrics applied. Feedback from the review should be used in order to improve the evaluation process and evaluation modules. When it is necessary to improve the evaluation modules, the data collection for extra indicators should be included, in order to validate them for later use.

Note: Quality evaluation review and feedback is described in ISO/IEC 14598-2.

Annex A
(informative)

Definitions from other standards

Definitions are from ISO/IEC 14598-1 unless otherwise indicated.

A.1
acquirer
an organization that acquires or procures a system, software product or software service from a supplier

[ISO/IEC 12207: 1995]

A.2
attribute
a measurable physical or abstract property of an entity

NOTE Attributes can be internal or external.

A.3
developer
an organization that performs development activities (including requirements analysis, design, testing through acceptance) during the software life cycle process

[ISO/IEC 12207: 1995]

A.4
direct measure
a measure of an attribute that does not depend upon a measure of any other attribute

A.5
evaluation module
a package of evaluation technology for a specific software quality characteristic or subcharacteristic

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

A.6
external measure
an indirect measure of a product derived from measures of the behaviour of the system of which it is a part

NOTE 1 The system includes any associated hardware, software (either custom software or off-the-shelf software) and users.

NOTE 2 The number of failures found during testing is an external measure of the number of faults in the program because the number of failures are counted during the operation of a computer system running the program.

NOTE 3 External measures can be used to evaluate quality attributes closer to the ultimate objectives of the design.

A.7
external quality
the extent to which a product satisfies stated and implied needs when used under specified conditions

A.8
failure
the termination of the ability of a product to perform a required function or its inability to perform within previously specified limits

A.9
fault
an incorrect step, process or data definition in a computer program

NOTE This definition is taken from IEEE 610.12-1990.

A.10
implied needs
needs that may not have been stated but are actual needs when the entity is used in particular conditions

NOTE Implied needs are real needs which may not have been documented.

A.11
indicator
a measure that can be used to estimate or predict another measure

NOTE 1 The predicted measure may be of the same or a different software quality characteristic.

NOTE 2 Indicators may be used both to estimate software quality attributes and to estimate attributes of the development process. They are imprecise indirect measures of the attributes.

A.12
indirect measure
a measure of an attribute that is derived from measures of one or more other attributes

NOTE An external measure of an attribute of a computing system (such as the response time to user input) is an indirect measure of attributes of the software as the measure will be influenced by attributes of the computing environment as well as attributes of the software.

A.13
intermediate software product
a product of the software development process that is used as input to another stage of the software development process

NOTE In some cases an intermediate product may also be an end product.

A.14
internal measure
a measure of the product itself, either direct or indirect

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

A.15
internal quality
the totality of attributes of a product that determine its ability to satisfy stated and implied needs when used under specified conditions

NOTE 1 The term "internal quality", used in ISO/IEC 14598 to contrast with "external quality", has essentially the same meaning as "quality" in ISO 8402.

NOTE 2 The term "attribute" is used with the same meaning as the term "characteristic" used in 4.1.1, as the term "characteristic" is used in a more specific sense in ISO/IEC 9126.

A.16
maintainer
an organization that performs maintenance activities

[ISO/IEC 12207: 1995]

A.17
measure (verb)
make a measurement

A.18
measure (noun)
the number or category assigned to an attribute of an entity by making a measurement

A.19
measurement
the use of a metric to assign a value (which may be a number or category) from a scale to an attribute of an entity

NOTE Measurement can be qualitative when using categories. For example, some important attributes of software products, e.g. the language of a source program (ADA, C, COBOL, etc.) are qualitative categories.

A.20
metric
the defined measurement method and the measurement scale

NOTE 1 Metrics can be internal or external, and direct or indirect

NOTE 2 Metrics include methods for categorising qualitative data.

A.21
quality
the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs.

NOTE 1 In a contractual environment, or in a regulated environment, such as the nuclear safety field, needs are specified, whereas in other environments, implied needs should be identified and defined (ISO 8402 : 1994, note 1).

NOTE 2 In ISO/IEC 14598 the relevant entity is a software product.

[ISO 8402: 1994]

A.22
quality evaluation
systematic examination of the extent to which an entity is capable of fulfilling specified requirements

NOTE The requirements may be formally specified, as when a product is developed for a specific user under a contract, or specified by the development organization, 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.

[ISO 8402: 1994]

A.23
quality model
the set of characteristics and the relationships between them which provide the basis for specifying quality requirements and evaluating quality

A.24
quality in use
the extent to which a product used by specified users meets their needs to achieve specified goals with effectiveness, productivity, safety and satisfaction in specified contexts of use

A.25
rating
the action of mapping the measured value to the appropriate rating level. Used to determine the rating level associated with the software for a specific quality characteristic

A.26
rating level
a scale point on an ordinal scale which is used to categorise a measurement scale

NOTE 1 The rating level enables software to be classified (rated) in accordance with the stated or implied needs (see 10.2).

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

A.27
scale
a set of values with defined properties

NOTE Examples of 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 possess an absolute zero. Metrics using nominal or ordinal scales produce qualitative data, and metrics using interval and ratio scales produce quantitative data.

A.28
software
all or part of the programs, procedures, rules, and associated documentation of an information processing system

NOTE Software is an intellectual creation that is independent of the medium on which it is recorded.

[ISO/IEC 2382.1: 1993]

A.29
software product
the set of computer programs, procedures, and possibly associated documentation and data

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

[ISO/IEC 12207: 1995]

A.30
supplier
an organization 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.31
system
an integrated composite that consists of one or more of the processes, hardware, software, facilitires and people, that provides a capability to satisfy a stated need or objective

[ISO/IEC 12207: 1995]

A.32
user
an individual that uses the software product to perform a specific function

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

A.33
validation
confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled

NOTE 1 In design and development, validation concerns the process of examining a product to determine conformity with user needs.

NOTE 2 Validation is normally performed on the final product under defined operating conditions. It may be necessary in earlier stages.

NOTE 3 "Validated" is used to designate the corresponding status.

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

[ISO 8402: 1994]

A.34
verification
confirmation by examination and provision of objective evidence that specified requirements have been fulfilled

NOTE 1 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.

NOTE 2 "Verified" is used to designate the corresponding status.

[ISO 8402:1994]

Bibliography

[1] ISO/IEC 9126-2, *Information technology - Software product quality - External metrics.*

[2] ISO/IEC 9126-3, *Information technology - Software product quality - Internal metrics.*

[3] ISO/IEC 9126-4, *Information technology - Software product quality – Quality in use metrics.*

[4] ISO/IEC 12207: *Information technology - Software life cycle processes.*

[5] ISO/IEC 15026: *Information technology - System and software integrity levels.*

ICS 35.080

Price based on 16 pages