INTERNATIONAL STANDARD

ISO/IEC 9126-1

First edition
2001-06-15

Software engineering — Product quality —

Part 1: Quality model

Génie du logiciel — Qualité des produits —

Partie 1: Modèle de qualité

Reference number
ISO/IEC 9126-1:2001(E)

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.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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

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.

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 9126 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 9126-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software engineering*.

This first edition of ISO/IEC 9126-1, together with the other parts of ISO/IEC 9126, cancels and replaces ISO/IEC 9126:1991, which has been technically revised.

ISO/IEC 9126 consists of the following parts, under the general title *Software engineering — Product quality*:

— *Part 1: Quality model*

— *Part 2: External metrics*

— *Part 3: Internal metrics*

— *Part 4: Quality in use metrics*

Annex A forms a normative part of this part of ISO/IEC 9126-1. Annexes B and C are for information only.

Introduction

Computers are being used in an increasingly wide variety of application areas, and their correct operation is often critical for business success and/or human safety. Developing or selecting high quality software products is therefore of prime importance. Comprehensive specification and evaluation of software product quality is a key factor in ensuring adequate quality. This can be achieved by defining appropriate quality characteristics, taking account of the purpose of usage of the software product. It is important that every relevant software product quality characteristic is specified and evaluated, whenever possible using validated or widely accepted metrics.

ISO/IEC 9126 (1991): Software product evaluation - Quality characteristics and guidelines for their use, which was developed to support these needs, defined six quality characteristics and described a software product evaluation process model.

As quality characteristics and associated metrics can be useful not only for evaluating a software product but also for defining quality requirements and other usage, ISO/IEC 9126 (1991) has been replaced by two related multipart standards: ISO/IEC 9126 (Software product quality) and ISO/IEC 14598 (Software product evaluation). The software product quality characteristics defined in this part of ISO/IEC 9126 can be used to specify both functional and non-functional customer and user requirements.

This part of ISO/IEC 9126 is a revision of ISO/IEC 9126 (1991), and retains the same software quality characteristics. The major differences are:

- the introduction of normative subcharacteristics, most of which are based on the informative subcharacteristics in ISO/IEC 9126 (1991);
- the specification of a quality model;
- the introduction of quality in use;
- removal of the evaluation process (which is now specified in the ISO/IEC 14598 standards);
- co-ordination of the content with ISO/IEC 14598-1.

The relationship between the standards in the ISO/IEC 9126 and ISO/IEC 14598 series (see Annex D) is illustrated in Figure 1.

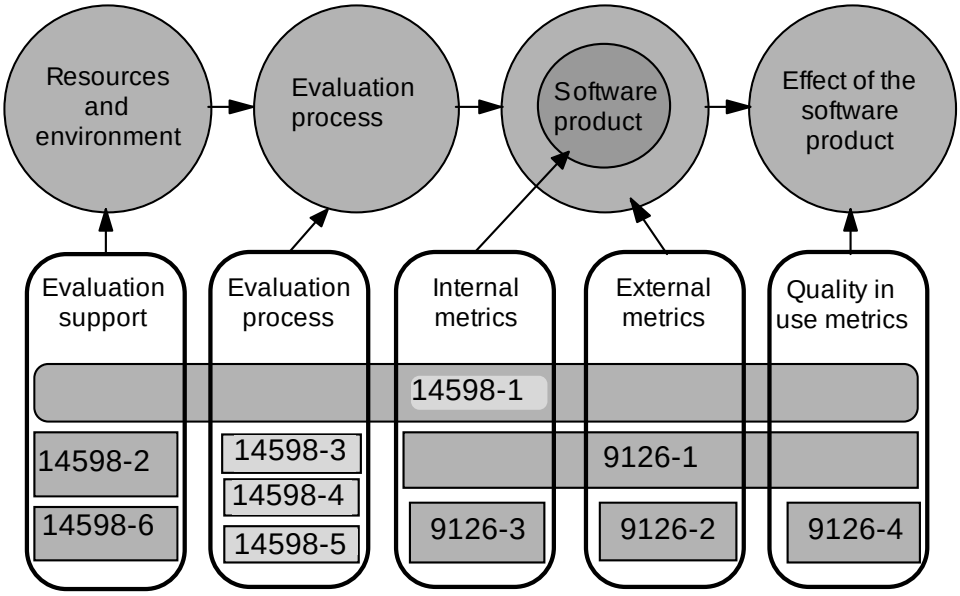


Figure 1 - Relationship between ISO/IEC 9126 and ISO/IEC 14598 standards

Software engineering — Product quality —

Part 1: Quality model

1 Scope

This part of ISO/IEC 9126 describes a two-part model for software product quality: a) internal quality and external quality, and b) quality in use. The first part of the model specifies six characteristics for internal and external quality, which are further subdivided into subcharacteristics. These subcharacteristics are manifested externally when the software is used as a part of a computer system, and are a result of internal software attributes. This part of ISO/IEC 9126 does not elaborate the model for internal and external quality below the level of subcharacteristics.

The second part of the model specifies four quality in use characteristics, but does not elaborate the model for quality in use below the level of characteristics. Quality in use is the combined effect for the user of the six software product quality characteristics.

The characteristics defined are applicable to every kind of software, including computer programs and data contained in firmware. The characteristics and subcharacteristics provide consistent terminology for software product quality. They also provide a framework for specifying quality requirements for software, and making trade-offs between software product capabilities.

Normative Annex A provides recommendations and requirements for software product metrics and quality in use metrics. Examples of these metrics are contained in other parts of ISO/IEC 9126. These metrics are applicable when specifying the quality requirements and the design goals for software products, including intermediate products. An explanation of how this quality model can be applied in software product evaluation is contained in ISO/IEC 14598-1.

This part of ISO/IEC 9126 enables software product quality to be specified and evaluated from different perspectives by those associated with acquisition, requirements, development, use, evaluation, support, maintenance, quality assurance and audit of software. It can for example be used by developers, acquirers, quality assurance staff and independent evaluators, particularly those responsible for specifying and evaluating software product quality. Examples of uses of the quality model defined in this part of ISO/IEC 9126 are to:

- validate the completeness of a requirements definition;
- identify software requirements;
- identify software design objectives;
- identify software testing objectives;
- identify quality assurance criteria;
- identify acceptance criteria for a completed software product.

NOTE 1 This part of ISO/IEC 9126 can be used in conjunction with ISO/IEC 15504 (which is concerned with the software process assessment) to provide:

- a framework for software product quality definition in the customer-supplier process;

- support for review, verification and validation, and a framework for quantitative quality evaluation, in the support process;
- support for setting organisational quality goals in the management process.

NOTE 2 This part of ISO/IEC 9126 can be used in conjunction with ISO/IEC 12207 (which is concerned with the software lifecycle) to provide:

- a framework for software product quality requirements definition in the primary lifecycle process;
- support for review, verification and validation in supporting lifecycle processes.

NOTE 3 This part of ISO/IEC 9126 can be used in conjunction with ISO 9001 (which is concerned with quality assurance processes) to provide:

- support for setting quality goals;
- support for design review, verification and validation.

2 Conformance

Any software product quality requirement, specification or evaluation that conforms to this part of ISO/IEC 9126 shall either use the characteristics and subcharacteristics from clauses 6 and 7, giving the reasons for any exclusions, or describe its own categorisation of software product quality attributes and provide a mapping to the characteristics and subcharacteristics in clauses 6 and 7.

A software product quality requirement or specification that contains metrics used for comparison shall state whether the metrics have the properties specified in A.4.

3 Normative reference

The following normative document contains provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 9126. 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 9126 are encouraged to investigate the possibility of applying the most recent edition of the normative document 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 14598-1:1999, *Information technology — Software product evaluation — Part 1: General overview.*

4 Terms and definitions

For the purposes of all parts of ISO/IEC 9126, the following definition and the definitions contained in ISO/IEC 14598-1 apply.

NOTE The definitions contained in ISO/IEC 14598-1 are reproduced in informative annex B.

4.1
level of performance
the degree to which the needs are satisfied, represented by a specific set of values for the quality characteristics

5 Quality model framework

This clause describes a quality model framework which explains the relationship between different approaches to quality. A specific implementation of this quality model is given in clauses 6 and 7.5.1.

5.1 Approaches to quality

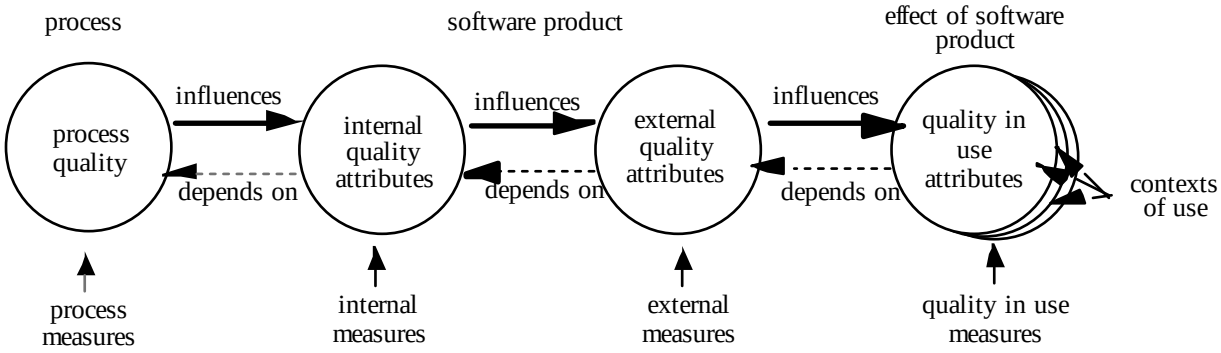


Figure 2 - Quality in the lifecycle

User quality needs include requirements for quality in use in specific contexts of use. These identified needs can be used when specifying external and internal quality using software product quality characteristics and subcharacteristics.

Evaluation of software products in order to satisfy software quality needs is one of the processes in the software development lifecycle. Software product quality can be evaluated by measuring internal attributes (typically static measures of intermediate products), or by measuring external attributes (typically by measuring the behaviour of the code when executed), or by measuring quality in use attributes. The objective is for the product to have the required effect in a particular context of use (Figure 2).

Process quality (the quality of any of the lifecycle processes defined in ISO/IEC 12207) contributes to improving product quality, and product quality contributes to improving quality in use. Therefore, assessing and improving a process is a means to improve product quality, and evaluating and improving product quality is one means of improving quality in use. Similarly, evaluating quality in use can provide feedback to improve a product, and evaluating a product can provide feedback to improve a process.

Appropriate internal attributes of the software are a pre-requisite for achieving the required external behaviour, and appropriate external behaviour is a pre-requisite for achieving quality in use (Figure 2).

The requirements for software product quality will generally include assessment criteria for internal quality, external quality and quality in use, to meet the needs of developers, maintainers, acquirers and end users. (See ISO/IEC 14598-1:1999, clause 8.)

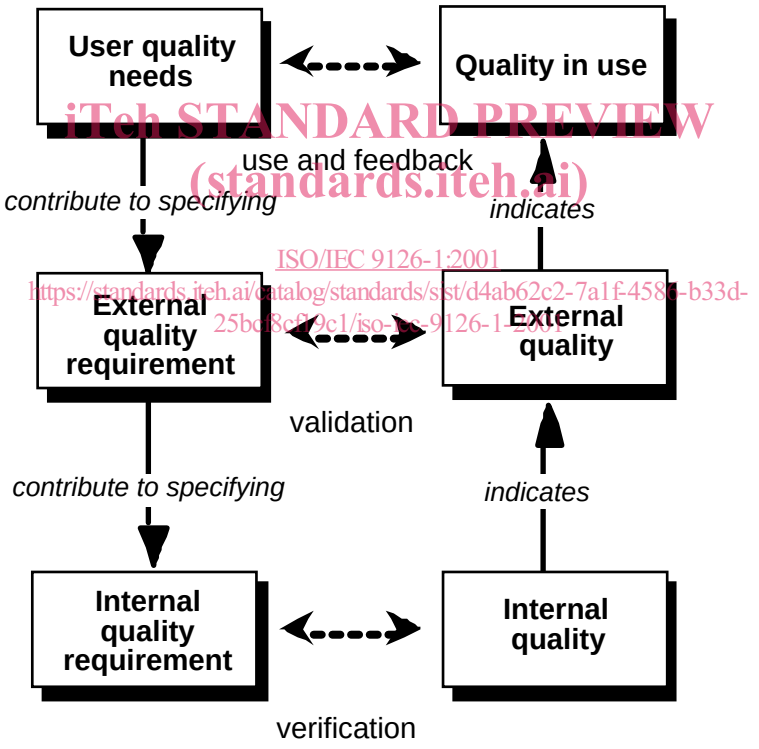
5.2 Product quality and the lifecycle

The views of internal quality, external quality and quality in use change during the software lifecycle. For example, quality specified as quality requirements at the start of the lifecycle is mostly seen from the external and users' view, and it differs from the interim product quality, such as design quality, which is mostly seen from the internal and developers view. The technologies used for achieving the necessary level of quality, such as specification and evaluation of quality, need to support these diverse points of view. It is necessary to define these perspectives and the associated technologies for quality, in order to manage quality properly at each stage of the lifecycle.

The goal is to achieve the necessary and sufficient quality to meet the real needs of users. ISO 8402 defines quality in terms of the ability to satisfy stated and implied needs. However, needs stated by a user do not always reflect the real user needs, because: (1) a user is often not aware of his real needs, (2) needs may change after they are stated, (3) different users may have different operating environments, and (4) it may be impossible to consult all the possible types of user, particularly for off-the-shelf software. So quality requirements cannot be completely defined before the beginning of design. Yet, it is necessary to understand the real user needs in as much detail as possible, and represent these in the requirements. The goal is not necessarily to achieve perfect quality, but the necessary and sufficient quality for each specified context of use when the product is delivered and actually used by users.

Measurement scales for the metrics used for quality requirements can be divided into categories corresponding to different degrees of satisfaction of the requirements. For example, the scale could be divided into two categories: unsatisfactory and satisfactory, or into four categories: exceeds requirements, target, minimally acceptable and unacceptable (see ISO/IEC 14598-1). The categories should be specified so that both the user and the developer can avoid unnecessary cost and schedule overruns.

There are different views of product quality and associated metrics at different stages in the software lifecycle (see Figure 3).



NOTE This figure is a simplified version of ISO/IEC 14598-1:1999 Figure 4, modified to be consistent with ISO/IEC 9126-1.

Figure 3 - Quality in the software lifecycle

User quality needs can be specified as quality requirements by quality in use metrics, by external metrics, and sometimes by internal metrics. These requirements specified by metrics should be used as criteria when a product is validated. Achieving a product which satisfies the user's needs normally requires an iterative approach to software development with continual feedback from a user perspective.

NOTE Guidance on design processes for interactive systems is given in ISO 13407.

External Quality Requirements specify the required level of quality from the external view. They include requirements derived from user quality needs, including quality in use requirements. External quality requirements are used as the target for validation at various stages of development. External quality requirements for all the quality characteristics defined in this part of ISO/IEC 9126 should be stated in the quality requirements specification using external metrics, should be transformed into internal quality requirements, and should be used as criteria when a product is evaluated.

Internal Quality Requirements specify the level of required quality from the internal view of the product. Internal quality requirements are used to specify properties of interim products. These can include static and dynamic models, other documents and source code. Internal quality requirements can be used as targets for validation at various stages of development. They can also be used for defining strategies of development and criteria for evaluation and verification during development. This may include the use of additional metrics (e.g. for reusability) which are outside the scope of ISO/IEC 9126. Specific internal quality requirements should be specified quantitatively using internal metrics.

Internal quality is the totality of characteristics of the software product from an internal view. Internal quality is measured and evaluated against the internal quality requirements. Details of software product quality can be improved during code implementation, reviewing and testing, but the fundamental nature of the software product quality represented by internal quality remains unchanged unless redesigned.

Estimated (or Predicted) External Quality is the quality that is estimated or predicted for the end software product at each stage of development for each quality characteristic, based on knowledge of the internal quality.

External Quality is the totality of characteristics of the software product from an external view. It is the quality when the software is executed, which is typically measured and evaluated while testing in a simulated environment with simulated data using external metrics. During testing, most faults should be discovered and eliminated. However, some faults may still remain after testing. As it is difficult to correct the software architecture or other fundamental design aspects of the software, the fundamental design usually remains unchanged throughout testing.

Estimated (or Predicted) Quality in Use is the quality that is estimated or predicted for the end software product at each stage of development for each quality in use characteristic, and based on knowledge of the internal and external quality.

NOTE External quality and quality in use can be estimated and predicted during development for each quality characteristic defined in this part of ISO/IEC 9126 when proper technologies are developed. However as the current state of the art does not provide all the support necessary for the purposes of prediction, more technology should be developed to show the co-relation between internal quality external quality and quality in use.

Quality in Use is the user's view of the quality of the software product when it is used in a specific environment and a specific context of use. It measures the extent to which users can achieve their goals in a particular environment, rather than measuring the properties of the software itself (quality in use is defined in clause 7).

NOTE 'Users' refers to any type of intended users, including both operators and maintainers, and their requirements can be different.

The level of quality in the users' environment may be different from that in the developers' environment, because of differences between the needs and capabilities of different users and differences between different hardware and support environments. The user evaluates only those attributes of software, which are used for his tasks. Sometimes, software attributes specified by an end user during the requirements analysis phase, no longer meet the user requirements when the product is in use, because of changing user requirements and the difficulty of specifying implied needs.

5.3 Items to be evaluated

Items can be evaluated by direct measurement, or indirectly by measuring their consequences. For example, a process may be assessed indirectly by measuring and evaluating it's product, and a product may be evaluated indirectly by measuring the task performance of a user (using quality in use metrics).

Software never runs alone, but always as part of a larger system typically consisting of other software products with which it has interfaces, hardware, human operators, and workflows. The completed software product can be evaluated by the levels of the chosen external metrics. These metrics describe its interaction with its environment, and are assessed by observing the software in operation. Quality in use can be measured by the extent to which a product used by specified users meets their needs to achieve specified goals with effectiveness, productivity, safety and satisfaction. This will normally be complemented by measures of more specific software product quality characteristics, which is also possible earlier in the development process.

At the earliest stages of development, only resources and process can be measured. When intermediate products (specifications, source code, etc.) become available, these can be evaluated by the levels of the chosen internal metrics. These metrics can be used to predict values of the external metrics. They may also be measured in their own right, as essential pre-requisites for external quality.

A further distinction can be made between the evaluation of a software product and the evaluation of the system in which it is executed.

NOTE 1 For example, the reliability of a system is assessed by observing all failures due to whatever cause (hardware, software, human error, etc.), whereas the reliability of the software product is assessed by extracting from the observed failures only those that are due to faults (originating from requirements, design or implementation) in the software.

Also, where the boundary of the system is judged to be, depends upon the purpose of the evaluation, and upon who the users are.

NOTE 2 For example, if the users of an aircraft with a computer-based flight control system are taken to be the passengers, then the system upon which they depend includes the flight crew, the airframe, and the hardware and software in the flight control system, whereas if the flight crew are taken to be the users, then the system upon which they depend consists only of the airframe and the flight control system.

5.4 Using a quality model

Software product quality should be evaluated using a defined quality model. The quality model should be used when setting quality goals for software products and intermediate products. Software product quality should be hierarchically decomposed into a quality model composed of characteristics and subcharacteristics which can be used as a checklist of issues related to quality. Clauses 6 and 7 define a hierarchical quality model (although other ways of categorising quality may be more appropriate in particular circumstances).

It is not practically possible to measure all internal and external subcharacteristics for all parts of a large software product. Similarly it is not usually practical to measure quality in use for all possible user-task scenarios. Resources for evaluation need to be allocated between the different types of measurement dependent on the business objectives and the nature of the product and design processes.