

Australian/New Zealand Standard™

Software engineering—Product quality

Part 1: Quality model



AS/NZS ISO/IEC 9126.1:2005

This Joint Australian/New Zealand Standard was prepared by Joint Technical Committee IT-015, Software and Systems Engineering. It was approved on behalf of the Council of Standards Australia on 4 April 2005 and on behalf of the Council of Standards New Zealand on 18 March 2005.
This Standard was published on 29 April 2005.

The following are represented on Committee IT-015:

Australian Computer Society
Australian Information Industry Association
Australian Society for Technical Communication (NSW)
Australian Software Metrics Association
Department of Defence (Australia)
Griffith University
New Zealand Organisation for Quality
Software Quality Association (ACT)
Software Quality Association (NSW)
Systems Engineering Society of Australia
The University of New South Wales
The University of Queensland
University of South Australia
University of Technology, Sydney

Keeping Standards up-to-date

Standards are living documents which reflect progress in science, technology and systems. To maintain their currency, all Standards are periodically reviewed, and new editions are published. Between editions, amendments may be issued. Standards may also be withdrawn. It is important that readers assure themselves they are using a current Standard, which should include any amendments which may have been published since the Standard was purchased.

Detailed information about joint Australian/New Zealand Standards can be found by visiting the Standards Web Shop at www.standards.com.au or Standards New Zealand web site at www.standards.co.nz and looking up the relevant Standard in the on-line catalogue.

Alternatively, both organizations publish an annual printed Catalogue with full details of all current Standards. For more frequent listings or notification of revisions, amendments and withdrawals, Standards Australia and Standards New Zealand offer a number of update options. For information about these services, users should contact their respective national Standards organization.

We also welcome suggestions for improvement in our Standards, and especially encourage readers to notify us immediately of any apparent inaccuracies or ambiguities. Please address your comments to the Chief Executive of either Standards Australia or Standards New Zealand at the address shown on the back cover.

This Standard was issued in draft form for comment as DR 05038.

Australian/New Zealand Standard™

Software engineering—Product quality

Part 1: Quality model

Originated as AS/NZS 4216:1994.
Revised and redesignated AS/NZS ISO/IEC 9126.1:2005.

COPYRIGHT

© Standards Australia/Standards New Zealand

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher.

Jointly published by Standards Australia, GPO Box 5420, Sydney, NSW 2001 and Standards New Zealand, Private Bag 2439, Wellington 6020

PREFACE

This Standard was prepared by the Joint Standards Australia/Standards New Zealand Committee IT-015, Software and Systems Engineering, to supersede AS/NZS 4216:1994, *Information technology—Software product evaluation—Quality characteristics and guidelines for their use*.

This Standard is identical with, and has been reproduced from ISO/IEC 9126-1:2001, *Software engineering—Product quality, Part 1: Quality model*.

The objective of this Standard is to provide the user with information that 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.

This Standard is Part 1 of AS/NZS ISO/IEC 9126, *Software engineering—Product quality*, which is published in parts as follows:

Part 1: Quality model (this Standard)

Part 2: External metrics

Part 3: Internal metrics

Part 4: Quality in use metrics

The term ‘normative’ has been used in this Standard to define the application of the annex to which it applies. A ‘normative’ annex is an integral part of a Standard.

As this Standard is reproduced from an international standard, the following applies:

- (a) Its number appears on the cover and title page while the international standard number appears only on the cover
- (b) In the source text ‘this International Standard’ should read ‘this Australian/New Zealand Standard’.
- (c) A full point substitutes for a comma when referring to a decimal marker.

References to International Standards should be replaced by references to Australian or Australian/New Zealand Standards, as follows:

<i>Reference to International Standard</i>		<i>Australian/New Zealand Standard</i>	
ISO/IEC		AS/NZS	
14598	Information technology—Software product evaluation	14598	Information technology—Software product evaluation
14598-1	Part 1: General overview	14598.1	Part 1: General overview

CONTENTS

Page

1 Scope	1
2 Conformance.....	2
3 Normative reference	2
4 Terms and definitions.....	2
5 Quality model framework.....	3
5.1 Approaches to quality.....	3
5.2 Product quality and the lifecycle.....	3
5.3 Items to be evaluated	6
5.4 Using a quality model.....	6
6 Quality model for external and internal quality	7
6.1 Functionality	7
6.2 Reliability	8
6.3 Usability	9
6.4 Efficiency	10
6.5 Maintainability.....	10
6.6 Portability.....	11
7 Quality model for quality in use	12
7.1 Quality in use.....	12
Annex A (normative) Metrics	14
A.1 Software metrics	14
A.2 Quality in use metrics	15
A.3 Choice of metrics and measurement criteria	16
A.4 Metrics used for comparison.....	16
Annex B (informative) Definitions from other standards	18
Annex C (informative) History of the work.....	23
Bibliography.....	25

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.

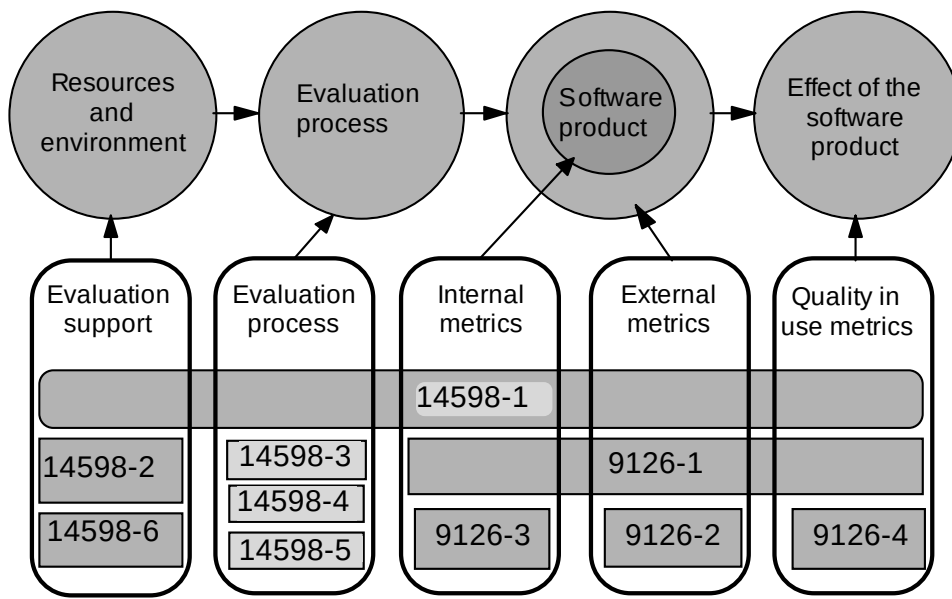


Figure 1 - Relationship between ISO/IEC 9126 and ISO/IEC 14598 standards

NOTES

AUSTRALIAN/NEW ZEALAND STANDARD

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

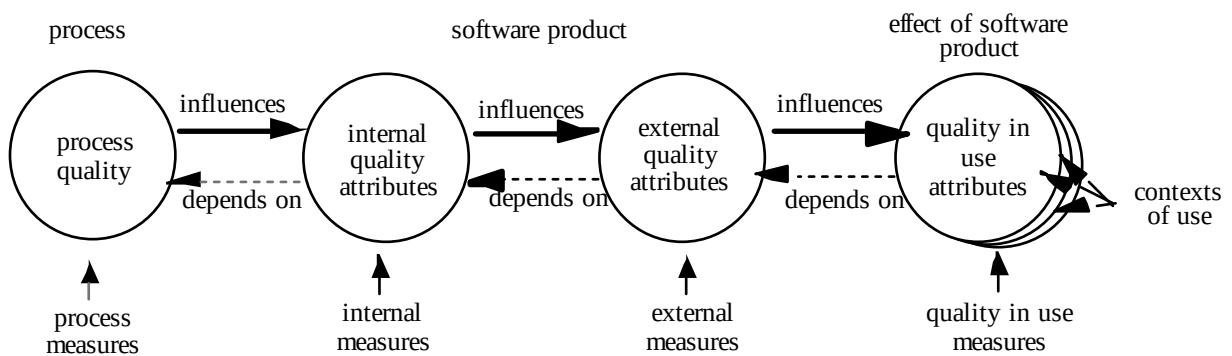


Figure 2 - Quality in the lifecycle

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.)

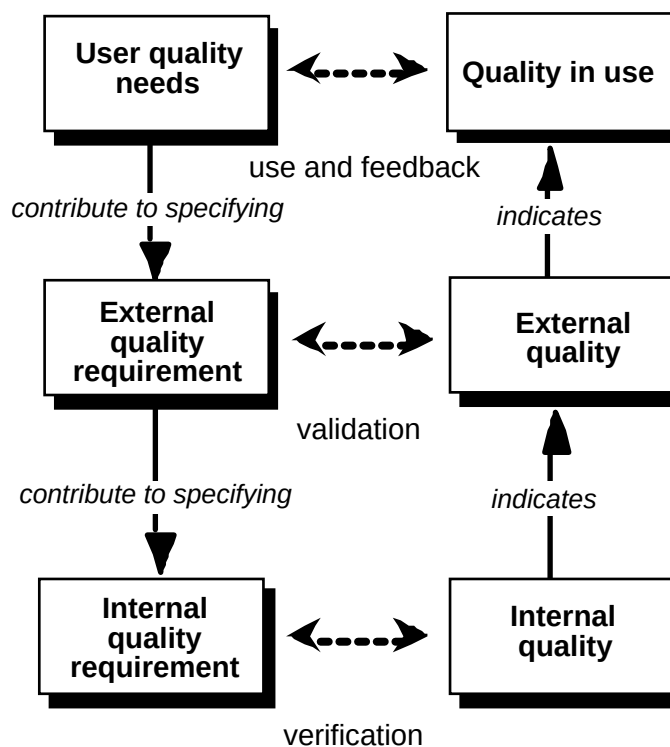
5.2 Product quality and the lifecycle

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.