

Australian/New Zealand Standard™

**Software engineering—Software
product Quality Requirements and
Evaluation (SQuaRE)—Guide to SQuaRE**



AS/NZS ISO/IEC 25000:2007

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 3 November 2006 and on behalf of the Council of Standards New Zealand on 17 November 2006.
This Standard was published on 8 January 2007.

The following are represented on Committee IT-015:

Australian Computer Society
Australian Electrical and Electronic Manufacturers Association
Australian Society of Technical Communications
Australian Software Metrics Association
Engineers Australia/ACTS Joint Board in Software Engineering
Griffith University
National Association of Testing Authorities Australia
National ICT Australia
New Zealand Organisation for Quality
Software Quality Association, ACT
Software Quality Association, NSW
Systems Engineering Society of Australia
The University of Queensland
University of Auckland, NZ
University of South Australia
University of Technology, Sydney
Vendor Interests, NZ

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 Standards may also be withdrawn.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 06547.

Australian/New Zealand Standard™

**Software engineering—Software
product Quality Requirements and
Evaluation (SQuaRE)—Guide to SQuaRE**

First published as AS/NZS ISO/IEC 25000:2007.

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 476, Sydney, NSW 2001 and Standards New Zealand, Private Bag 2439, Wellington 6020

ISBN 0 7337 7965 4

PREFACE

This Standard was prepared by the Joint Standards Australia/Standards New Zealand Committee IT-015, Software and Systems Engineering.

The objective of this Standard is to provide Software Engineers with guidance for the use of the new series of International Standards named Software product Quality Requirements and Evaluation (SQuaRE). This guide provides a general overview of SQuaRE contents, common reference models and definitions, as well as the relationship among the documents, allowing users of the guide a good understanding of those series of standards, according to their purpose of use.

This Standard is identical with, and has been reproduced from ISO/IEC 25000:2005, *Software engineering—Software product Quality Requirements and Evaluation (SQuaRE)—Guide to SQuaRE*.

Many of the documents listed in the Bibliography have been adopted as Australian or Australian/New Zealand Standards.

For further information, refer to the Standards Australia catalogue or website.

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:

The term ‘informative’ has been used in this Standard to define the application of the annex to which it applies. An ‘informative’ annex is only for information and guidance.

CONTENTS

Page

1	Scope	1
2	Conformance	1
3	Normative references	1
4	Terms and definitions.....	1
5	SQuaRE: Software product Quality Requirements and Evaluation – the series of standards on product quality requirements and evaluation	10
5.1	Organisation of SQuaRE series of standards.....	10
5.2	SQuaRE: overview of documents within series	11
5.3	SQuaRE common models	12
Annex A (informative)	Relationship between SQuaRE series and other ISO Standards.....	16
A.1	ISO/IEC 12207:1995/Amd 1:2002	16
A.2	ISO/IEC 15504.....	16
A.3	ISO 9000 family of standards.....	16
A.4	ISO/IEC 15939.....	19
A.5	ISO/IEC 15288.....	19
Annex B (informative)	Overview of ISO/IEC 14598 and ISO/IEC 9126.....	21
B.1	Overview of ISO/IEC 14598 and ISO/IEC 9126	21
B.2	Quality model framework	22
B.3	Evaluation process	23
B.4	Support for evaluation.....	24
B.5	Software quality characteristics and metrics	24
B.6	The evaluation process	26
Annex C (informative)	History and transition process between ISO/IEC 9126, ISO/IEC 14598 and SQuaRE series of standards	34
C.1	History.....	34
C.2	Relationship between ISO/IEC 9126 and ISO/IEC 14598 series and SQuaRE series of standards	35
Annex D (informative)	Examples of the application of SQuaRE series of standards	37
Bibliography		40

INTRODUCTION

Computers are being used in an increasingly wide variety of application areas, and their intended and 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, while taking account of the intended use 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 measures.

As quality characteristics and associated measures can be useful not only for evaluating a software product but also for defining quality requirements, the predecessor of SQuaRE, ISO/IEC 9126:1991 has been replaced by two related multipart International Standards: ISO/IEC 9126 (Software product quality) and ISO/IEC 14598 (Software product evaluation). The following points derived from practical use of both series gave the logical impulse for creating the new SQuaRE series of International Standards:

- Both ISO/IEC 9126 and ISO/IEC 14598 have common normative, referential and functional roots,
- ISO/IEC 9126 and ISO/IEC 14598 form a complementary set of standards,
- The independent life cycles of both series have created inconsistencies between them.

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

It has to be stressed that the SQuaRE series of International Standards is dedicated to software product quality only. SQuaRE ISO/IEC 25000n — Quality Management Division addresses software product quality requirements specification, measurement and evaluation, and is separate and distinct from the "Quality Management" of processes, which is defined in the ISO 9000 family of standards.

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

- the coordination of guidance on software product quality measurement and evaluation,
- guidance for the specification of software product quality requirements, and
- harmonization with ISO/IEC 15939 in the form of Software product Quality Measurement Reference Model presented in ISO/IEC 25020 - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) Measurement reference model and guide.

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

- the introduction of the new general reference model,
- the introduction of dedicated, detailed guides for each division,

- the introduction of Quality Measure elements within Quality Measurement Division,
- the introduction of the Quality Requirements Division,
- incorporation and revision of the evaluation processes,
- the introduction of guidance of practical use in form of examples,
- coordination and harmonization of the content with ISO/IEC 15939.

SQuaRE consists of the following five divisions:

- ISO/IEC 2500n - Quality Management Division,
- ISO/IEC 2501n - Quality Model Division,
- ISO/IEC 2502n - Quality Measurement Division,
- ISO/IEC 2503n - Quality Requirements Division, and
- ISO/IEC 2504n - Quality Evaluation Division,

ISO/IEC 25050 to ISO/IEC 25099 are reserved to be used for SQuaRE extension International Standards and/or Technical Reports.

SQuaRE provides:

- Terms and definitions,
- Reference models,
- General guide,
- Individual division guides, and
- International Standards for requirements specification, planning and management, measurement and evaluation purposes.

SQuaRE includes International Standards on quality model and measures, as well as on quality requirements and evaluation.

SQuaRE replaces the current ISO/IEC 9126 series and the 14598 series.

This part of SQuaRE series of standards is a new International Standard with the goal of providing a common set of reference models, terminology, definitions and guidance for practical use of the associated standards and technical reports.

NOTES

AUSTRALIAN/NEW ZEALAND STANDARD

Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE

1 Scope

This International Standard provides guidance for the use of the new series of International Standards named Software product Quality Requirements and Evaluation (SQuaRE). The purpose of this Guide is to provide a general overview of SQuaRE contents, common reference models and definitions, as well as the relationship among the documents, allowing users of the Guide a good understanding of those series of standards, according to their purpose of use. This document contains an explanation of the transition process between the old ISO/IEC 9126 and the 14598 series and SQuaRE and also presents information on how to use the ISO/IEC 9126 and 14598 series in their previous form.

SQuaRE series of standards is intended for, but not limited to, developers, acquirers and independent evaluators of software products, particularly those responsible for defining software quality requirements and for software product evaluation. It is recommended that users of the SQuaRE as well as ISO/IEC 14598 and 9126 series of standards also use this International Standard as a guide to execute their tasks.

2 Conformance

There is no particular conformance clause for this document. Users, for their intended use of SQuaRE series of Standards should follow individual conformance clauses stated in each document of the series.

3 Normative references

This International Standard does not require any normative references. All informative references are presented in the Bibliography.

4 Terms and definitions

For the purposes of this document, the following definitions apply.

NOTE The definitions are common to all parts of SQuaRE series of standards.

4.1

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.

4.2

analysis model

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

4.3**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.

4.4**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 quality measure elements.

4.5**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].

4.6**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

4.7**context of use**

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

[ISO 9241-11:1998]

4.8**custom software**

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

4.9**data**

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

[ISO/IEC 15939:2002]

4.10**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]

4.11**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.

4.12**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.

4.13**division of standards**

division forms a family of standards serving complementary purposes

4.14**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.

4.15**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]

4.16**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

4.17**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.

4.18**evaluator**

individual or organisation that performs an evaluation

4.19**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.

4.20

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.

4.21

fault

incorrect step, process or data definition in a computer program

[IEEE 610.12-1990]

4.22

functional requirement

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

[IEEE 610.12-1990]

NOTE The software 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]).

4.23

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.

4.24

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".

4.25

information need

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

[ISO/IEC 15939:2002]

4.26

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]

4.27**information system needs**

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

4.28**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.

4.29**intermediate software product needs**

needs that can be specified as quality requirements by internal measures

4.30**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.

4.31**maintainer**

individual or organisation that performs maintenance activities

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

4.32**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]

4.33**measure (verb)**

make a measurement

[ISO/IEC 14598-1:1999]

4.34**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.).

4.35**measurement function**

algorithm or calculation performed to combine two or more base measures

[ISO/IEC 15939:2002]

4.36**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].

4.37**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]

4.38**measurement process**

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

[ISO/IEC 15939:2002]

4.39**observation**

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

[ISO/IEC 15939:2002]

4.40**operator**

individual or organisation that operates the system

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

4.41**process**

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

[ISO 9000:2000]

4.42**quality in use (measure)**

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

4.43**quality measure elements**

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.

4.44**quality model**

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

4.45**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

4.46**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'.

4.47**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.

4.48**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.

4.49**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 standards software quality has the same meaning as software product quality.

4.50**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

4.51**software quality**

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

NOTE This definitions 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.

4.52

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.

4.53

software quality evaluation

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

4.54

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.

4.55

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].

4.56

stakeholder

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

4.57

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]

4.58

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]