

Australian/New Zealand Standard™

Software engineering—Product quality

Part 3: Internal metrics



AS/NZS ISO/IEC 9126.3:2005

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 Standards may also be withdrawn.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 05040.

STANDARDS AUSTRALIA/STANDARDS NEW ZEALAND

RECONFIRMATION
OF
AS/NZS ISO/IEC 9126.3:2005
Software engineering—Product quality
Part 3: Internal metrics

RECONFIRMATION NOTICE

Technical Committee IT-015 has reviewed the content of this publication and in accordance with Standards Australia procedures for reconfirmation, it has been determined that the publication is still valid and does not require change.

Certain documents referenced in the publication may have been amended since the original date of publication. Users are advised to ensure that they are using the latest versions of such documents as appropriate, unless advised otherwise in this Reconfirmation Notice.

Approved for reconfirmation in accordance with Standards Australia procedures for reconfirmation on 30 October 2015.

Approved for reconfirmation in New Zealand on behalf of the Standards Council of New Zealand on 18 May 2016.

The following are represented on Technical Committee IT-015:

Australian Computer Society
Australian Society for Technical Communication (NSW)
Charles Sturt University
Department of Defence (Australian Government)
Griffith University
Institute of IT Professionals New Zealand
IT Service Management Forum Australia
La Trobe University
National Association of Testing Authorities Australia
NEHTA
New Zealand Organisation for Quality
NSW Business Chamber
Quantitative Enterprise Software Performance
Systems Engineering Society of Australia
University of Technology, Sydney

NOTES

Australian/New Zealand Standard™

Software engineering—Product quality

Part 3: Internal metrics

First published as AS/NZS ISO/IEC 9126.3:2005.

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

ISBN 0 7337 6664 1



Documents

Document Sharing Hub

PREFACE

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

This Standard is identical with, and has been reproduced from, ISO/IEC TR 9126-3:2003, *Software engineering—Product quality. Part 3: Internal metrics*.

The objective of this Standard is to provide the user with internal metrics for quantitatively measuring external software quality in terms of characteristics and subcharacteristics defined in AS/NZS ISO/IEC 9126.1. It is intended to be used together with AS/NZS ISO/IEC 9126.1.

This Standard is Part 3 of AS/NZS ISO/IEC TR 9126, *Software engineering—Product quality*, which is published in parts as follows:

Part 1: Quality model

Part 2: External metrics

Part 3: Internal metrics (this Standard)

Part 4: Quality in use metrics

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.

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 Technical Report’ 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
14598-2	Part 2: Planning and management	14598.2	Part 2: Planning and management
14598-3	Part 3: Process for developers	14598.3	Part 3: Process for developers
14598-4	Part 4: Process for acquirers	14598.4	Part 4: Process for acquirers
14598-5	Part 5: Process for evaluators	14598.5	Part 5: Process for evaluators
14598-6	Part 6: Documentation of evaluation modules	14598.6	Part 6: Documentation of evaluation modules
ISO/IEC		AS/NZS ISO/IEC	
12207	Information technology—Software life cycle processes	12207	Information technology—Software life cycle processes

ISO/IEC		AS/NZS	
14143	Information technology—Software measurement—Functional size measurement	14143	Information technology—Software measurement—Functional size measurement
14143-1	Part 1: Definition of concepts	14143.1	Part 1: Definition of concepts
ISO/IEC		AS	
2382	Information technology—Vocabulary	1189	Information technology—Vocabulary
2382-20	Part 20: System development	1189.20	Part 20: System development

Only international references that have been adopted as Australian or Australian/New Zealand Standards have been listed.

CONTENTS

Page

1	Scope.....	1
2	Conformance	2
3	Normative references.....	2
4	Terms and definitions	2
5	Abbreviated terms	2
6	Use of software quality metrics	3
7	How to read and use the metrics tables	4
8	Metrics tables.....	4
8.1	Functionality metrics.....	4
8.1.1	Suitability metrics	5
8.1.2	Accuracy metrics	5
8.1.3	Interoperability metrics.....	5
8.1.4	Security metrics.....	5
8.1.5	Functionality compliance metrics	5
8.2	Reliability metrics	12
8.2.1	Maturity metrics.....	12
8.2.2	Fault tolerance metrics	12
8.2.3	Recoverability metrics	12
8.2.4	Reliability compliance metrics.....	12
8.3	Usability Metrics	17
8.3.1	Understandability metrics	17
8.3.2	Learnability metrics	17
8.3.3	Operability metrics	17
8.3.4	Attractiveness metrics	17
8.3.5	Usability compliance metrics.....	17
8.4	Efficiency metrics	24
8.4.1	Time behaviour metrics	24
8.4.2	Resource utilization metrics	24
8.4.3	Efficiency compliance metrics.....	24
8.5	Maintainability metrics	28
8.5.1	Analysability metrics	28
8.5.2	Changeability metrics	28
8.5.3	Stability metrics.....	28
8.5.4	Testability metrics	28
8.5.5	Maintainability compliance metrics.....	28
8.6	Portability metrics.....	34
8.6.1	Adaptability metrics	34
8.6.2	Installability metrics	34

8.6.3	Co-existence metrics.....	34
8.6.4	Replaceability metrics.....	34
8.6.5	Portability compliance metrics.....	34
Annex A (informative) Considerations When Using Metrics		41
A.1	Interpretation of measures.....	41
A.1.1	Potential differences between test and operational contexts of use	41
A.1.2	Issues affecting validity of results	42
A.1.3	Balance of measurement resources	42
A.1.4	Correctness of specification.....	42
A.2	Validation of Metrics	42
A.2.1	Desirable Properties for Metrics	42
A.2.2	Demonstrating the Validity of Metrics	43
A.3	Use of metrics for estimation (judgement) and prediction (forecast)	44
A.3.1	Quality characteristics prediction by current data.....	44
A.3.2	Current quality characteristics estimation on current facts	44
A.4	Detecting deviations and anomalies in quality problem prone components	45
A.5	Displaying measurement results.....	45
Annex B (informative) Use of Quality in Use, External & Internal Metrics (Framework Example)		46
B.1	Introduction	46
B.2	Overview of Development and Quality Process	46
B.3	Quality Approach Steps	47
B.3.1	General.....	47
B.3.2	Step #1 Quality requirements identification	47
B.3.3	Step #2 Specification of the evaluation	48
B.3.4	Step #3 Design of the evaluation	50
B.3.5	Step #4 Execution of the evaluation	50
B.3.6	Step #5 Feedback to the organization	50
Annex C (informative) Detailed explanation of metric scale types and measurement types		51
C.1	Metric Scale Types	51
C.2	Measurement Types.....	52
C.2.1	Size Measure Type.....	52
C.2.2	Time measure type	55
C.2.2.0	General.....	55
C.2.3	Count measure type	56
Annex D (informative) Term(s).....		58
D.1	Definitions	58
D.1.1	Quality.....	58
D.1.2	Software and user.....	58
D.1.3	Measurement.....	59
Annex E (informative) Pure Internal Metrics		60
E.1	Pure Internal Metrics.....	60

Table 8.1.1 Suitability metrics	6
Table 8.1.2 Accuracy metrics	8
Table 8.1.3 Interoperability metrics	9
Table 8.1.4 Security metrics	10
Table 8.1.5 Functionality compliance metrics	11
Table 8.2.1 Maturity metrics	13
Table 8.2.2 Fault tolerance metrics	14
Table 8.2.3 Recoverability metrics	15
Table 8.2.4 Reliability compliance metrics	16
Table 8.3.1 Understandability metrics	18
Table 8.3.2 Learnability metrics	19
Table 8.3.3 Operability metrics	20
Table 8.3.4 Attractiveness metrics	22
Table 8.3.5 Usability compliance metrics	23
Table 8.4.1 Time behaviour metrics	25
Table 8.4.2 Resource utilisation metrics	26
Table 8.4.3 Efficiency compliance metrics	27
Table 8.5.1 Analysability metrics	29
Table 8.5.2 Changeability metrics	30
Table 8.5.3 Stability metrics	31
Table 8.5.4 Testability metrics	32
Table 8.5.5 Maintainability compliance metrics	33
Table 8.6.1 Adaptability metrics	35
Table 8.6.2 Installability metrics	37
Table 8.6.3 Co-existence metrics	38
Table 8.6.4 Replaceability metrics	39
Table 8.6.5 Portability compliance metrics	40
Table B.1 Quality Measurement Model	46
Table B.2 User Needs Characteristics & Weights	47
Table B.3 Quality measurement tables	48
Table B.4 Measurement plan	50

INTRODUCTION

This Technical Report provides internal metrics for measuring attributes of six external quality characteristics defined in ISO/IEC 9126-1. The metrics listed in this Technical Report are not intended to be an exhaustive set. Developers, evaluators, quality managers and acquirers may select metrics from this Technical Report for defining requirements, evaluating software products, measuring quality aspects and other purposes. They may also modify the metrics or use metrics which are not included here. This Technical Report is applicable to any kind of software product, although each of the metrics is not always applicable to every kind of software product.

ISO/IEC 9126-1 defines terms for the software quality characteristics and how these characteristics are decomposed into subcharacteristics. ISO/IEC 9126-1, however, does not describe how any of these subcharacteristics could be measured. ISO/IEC TR 9126-2 defines external metrics, ISO/IEC TR 9126-3 defines internal metrics and ISO/IEC 9126-4 defines quality in use metrics, for measurement of the characteristics or the subcharacteristics. Internal metrics measure the software itself, external metrics measure the behaviour of the computer-based system that includes the software, and quality in use metrics measure the effects of using the software in a specific context of use.

This Technical Report is intended to be used together with ISO/IEC 9126-1. It is strongly recommended to read ISO/IEC 14598-1 and ISO/IEC 9126-1, prior to using this Technical Report, particularly if the reader is not familiar with the use of software metrics for product specification and evaluation.

Clauses 1 to 7 and Annexes A to D are common to ISO/IEC TR 9126-2, ISO/IEC TR 9126-3, and ISO/IEC 9126-4. Annex E is for ISO/IEC TR 9126-3 use.

AUSTRALIAN/NEW ZEALAND STANDARD

Software engineering — Product quality —**Part 3:
Internal metrics****1 Scope**

This Technical Report defines internal metrics for quantitatively measuring external software quality in terms of characteristics and subcharacteristics defined in ISO/IEC 9126-1, and is intended to be used together with ISO/IEC 9126-1.

This Technical Report contains:

- I. an explanation of how to apply software quality metrics
- II. a basic set of metrics for each subcharacteristic
- III. an example of how to apply metrics during the software product life cycle

This Technical Report does not assign ranges of values of these metrics to rated levels or to grades of compliance, because these values are defined for each software product or a part of the software product, by its nature, depending on such factors as category of the software, integrity level and users' needs. Some attributes may have a desirable range of values, which does not depend on specific user needs but depends on generic factors; for example, human cognitive factors.

This Technical Report can be applied to any kind of software for any application. Users of this Technical Report can select or modify and apply metrics and measures from this Technical Report or may define application-specific metrics for their individual application domain. For example, the specific measurement of quality characteristics such as safety or security may be found in International Standards or Technical Reports provided by IEC 65 and ISO/IEC JTC 1/SC 27.

Intended users of this Technical Report include:

- Acquirer (an individual or organization that acquires or procures a system, software product or software service from a supplier);
- Evaluator (an individual or organization that performs an evaluation. An evaluator may, for example, be a testing laboratory, the quality department of a software development organization, a government organization or a user);
- Developer (an individual or organization that performs development activities, including requirements analysis, design, and testing through acceptance during the software life cycle process);
- Maintainer (an individual or organization that performs maintenance activities);
- Supplier (an individual or 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) when validating software quality at qualification test;
- User (an individual or organization that uses the software product to perform a specific function) when evaluating quality of software product at acceptance test;
- Quality manager (an individual or organization that performs a systematic examination of the software product or software services) when evaluating software quality as part of quality assurance and quality control.

2 Conformance

There are no conformance requirements in this Technical Report.

NOTE General conformance requirements for metrics are in ISO/IEC 9126-1 Quality model.

3 Normative references

ISO/IEC 9126-1:2001, *Software engineering — Product quality — Part 1: Quality model*

ISO/IEC TR 9126-2¹⁾, *Software engineering — Product quality — Part 2: External metrics*

ISO/IEC 9126-4¹⁾, *Software engineering — Product quality — Part 4: Quality in use metrics*

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

ISO/IEC 14598-2:2000, *Software engineering — Product evaluation — Part 2: Planning and management*

ISO/IEC 14598-3:2000, *Software engineering — Product evaluation — Part 3: Process for developers*

ISO/IEC 14598-4:1999, *Software engineering — Product evaluation — Part 4: Process for acquirers*

ISO/IEC 14598-5:1998, *Information technology — Software product evaluation — Part 5: Process for evaluators*

ISO/IEC 14598-6:2001, *Software engineering — Product evaluation — Part 6: Documentation of evaluation modules*

ISO/IEC 12207:1995, *Information technology — Software life cycle processes*

ISO/IEC 14143-1:1998, *Information technology — Software measurement — Functional size measurement — Part 1: Definition of concepts*

ISO 2382-20:1990, *Information technology — Vocabulary — Part 20: System development*

ISO 9241-10:1996, *Ergonomic requirements for office work with visual display terminals (VDTs) — Part 10: Dialogue principles*

4 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 14598-1:1999 and ISO/IEC 9126-1:2001 apply. They are also listed in Annex D.

5 Abbreviated terms

The following abbreviations are used in this Technical Report:

SQA — Software Quality Assurance (Group)

SLCP — Software Life Cycle Processes

1) To be published.

6 Use of software quality metrics

These Technical Reports (ISO/IEC TR 9126-2 External metrics, ISO/IEC TR 9126-3 Internal metrics and ISO/IEC 9126-4 Quality in use metrics) provide a suggested set of software quality metrics (external, internal and quality in use metrics) to be used with the ISO/IEC 9126-1 Quality model. The user of these Technical Reports may modify the metrics defined, and/or may also use metrics not listed. When using a modified or a new metric not identified in these Technical Reports, the user should specify how the metrics relate to the ISO/IEC 9126-1 quality model or any other substitute quality model that is being used.

The user of these Technical Reports should select the quality characteristics and subcharacteristics to be evaluated, from ISO/IEC 9126-1; identify the appropriate direct and indirect measures, identify the relevant metrics and then interpret the measurement result in an objective manner. The user of these Technical Reports also may select product quality evaluation processes during the software life cycle from the ISO/IEC 14598 series of standards. These give methods for measurement, assessment and evaluation of software product quality. They are intended for use by developers, acquirers and independent evaluators, particularly those responsible for software product evaluation (see Figure 1).

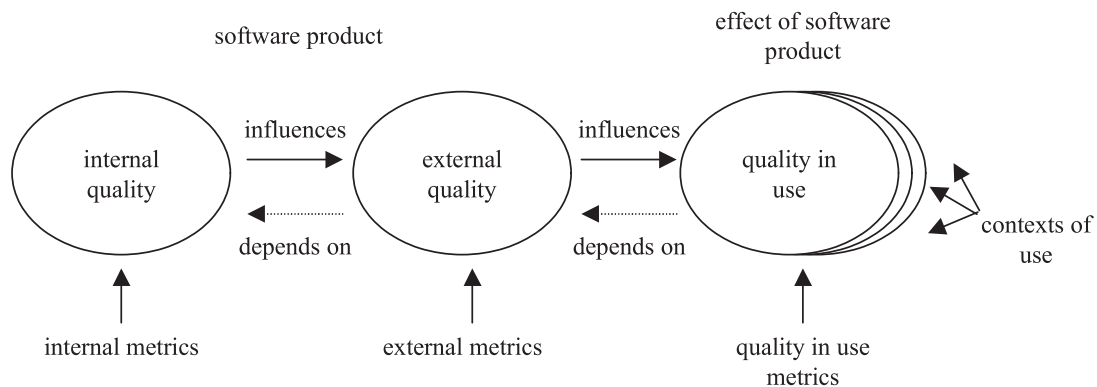


Figure 1 – Relationship between types of metrics

The internal metrics may be applied to a non-executable software product during its development stages (such as request for proposal, requirements definition, design specification or source code). Internal metrics provide the users with the ability to measure the quality of the intermediate deliverables and thereby predict the quality of the final product. This allows the user to identify quality issues and initiate corrective action as early as possible in the development life cycle.

The external metrics may be used to measure the quality of the software product by measuring the behaviour of the system of which it is a part. The external metrics can only be used during the testing stages of the life cycle process and during any operational stages. The measurement is performed when executing the software product in the system environment in which it is intended to operate.

The quality in use metrics measure whether a product meets the needs of specified users to achieve specified goals with effectiveness, productivity, safety and satisfaction in a specified context of use. This can be only achieved in a realistic system environment.

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 evaluated.

It is recommended to use internal metrics having a relationship as strong as possible with the target external metrics so that they can be used to predict the values of external metrics. However, it is often difficult to design a rigorous theoretical model that provides a strong relationship between internal metrics and external metrics. Therefore, a hypothetical model that may contain ambiguity may be designed and the extent of the relationship may be modelled statistically during the use of metrics.

Recommendations and requirements related to validity and reliability are given in ISO/IEC 9126-1, Clause A.4. Additional detailed considerations when using metrics are given in Annex A of this Technical Report.

7 How to read and use the metrics tables

The metrics listed in Clause 8 are categorized by the characteristics and subcharacteristics in ISO/IEC 9126-1. The following information is given for each metric in the table:

- a) **Metric name:** Corresponding metrics in the internal metrics table and external metrics table have similar names.
- b) **Purpose of the metric:** This is expressed as the question to be answered by the application of the metric.
- c) **Method of application:** Provides an outline of the application.
- d) **Measurement, formula and data element computations:** Provides the measurement formula and explains the meanings of the used data elements.

NOTE In some situations more than one formula is proposed for a metric.

- e) **Interpretation of measured value:** Provides the range and preferred values.
- f) **Metric scale type:** Type of scale used by the metric. Scale types used are; Nominal scale, Ordinal scale, Interval scale, Ratio scale and Absolute scale.

NOTE A more detailed explanation is given in Annex C.

- g) **Measure type:** Types used are; Size type (e.g. Function size, Source size), Time type (e.g. Elapsed time, User time), Count type (e.g. Number of changes, Number of failures).

NOTE A more detailed explanation is given in Annex C.

- h) **Input to measurement:** Source of data used in the measurement.
- i) **ISO/IEC 12207 SLCP Reference:** Identifies software life cycle process(es) where the metric is applicable.
- j) **Target audience:** Identifies the user(s) of the measurement results.

8 Metrics tables

The metrics listed in this clause are not intended to be an exhaustive set and may not have been validated. They are listed by software quality characteristics and subcharacteristics, in the order introduced in ISO/IEC 9126-1.

Metrics, which may be applicable, are not limited to these listed here. Additional specific metrics for particular purposes are provided in other related documents, such as functional size measurement or precise time efficiency measurement.

NOTE 1 It is recommended to refer a specific metric or measurement form from specific standards, technical reports or guidelines. Functional size measurement is defined in ISO/IEC 14143. An example of precise time efficiency measurement can be referred from ISO/IEC 14756.

Metrics should be validated before application in a specific environment (see Annex A).

NOTE 2 This list of metrics is not finalized, and may be revised in future versions of this Technical Report. Readers of this Technical Report are invited to provide feedback.

8.1 Functionality metrics

Internal functionality metrics are used for predicting if the software product in question will satisfy prescribed functional requirements and implied user needs.

8.1.1 Suitability metrics

Internal suitability metrics indicate a set of attributes for assessing explicitly functions to prescribed tasks, and for determining their adequacy for performing the tasks.

8.1.2 Accuracy metrics

Internal accuracy metrics indicate a set of attributes for assessing the capability of the software product to achieve correct or agreeable results.

8.1.3 Interoperability metrics

Internal Interoperability metrics indicate a set of attributes for assessing the capability of the software product's interaction with designated systems.

8.1.4 Security metrics

Internal security metrics indicate a set of attributes for assessing the capability of the software product to avoid illegal access to the system and/or data.

8.1.5 Functionality compliance metrics

Internal compliance metrics indicate a set of attributes for assessing the capability of the software product to comply to such items as standards, conventions or regulations of the user organization in relation to functionality.

Table 8.1.1 Suitability metrics

Internal suitability metrics									
Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value	Metric scale type	Measure type	Sources of input to measurement	ISO/IEC 12207 SLCP Reference	Target audience
Functional adequacy	How adequate are the checked functions?	Count the number of implemented functions that are suitable for performing the specified tasks, then measure the ratio of it to functions implemented. The following may be measured; -all or parts of design specifications -completed modules/parts of software products	$X=1-A/B$ A= Number of functions in which problems are detected in evaluation B= Number of functions checked	$0 \leq X \leq 1$ The closer to 1, the more adequate.	absolute	X=count/ count A=count B=count	Req spec Design Source code Review report	6.5 Validation 6.6 Joint review	Requirers Developers
Functional implementation completeness	How complete is the functional implementation?	Count the number of missing functions detected in evaluation and compare with the number of function described in the requirement specifications.	$X=1-A/B$ A=Number of missing functions detected in evaluation. B=Number of functions described in requirement specifications	$0 \leq X \leq 1$ The closer to 1, the more complete.	absolute	X=count/ count A=count B=count	Req spec Design Source code Review report	6.5 Validation 6.6 Joint review	Requirers Developers

FOOTNOTE

Input to the measurement process is the updated requirement specifications. Any changes identified during life cycle must be applied to the requirement specifications before using in measurement process.

Table 8.1.1 (continued)

Internal suitability metrics									
Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value	Metric scale type	Measure type	Sources of input to measurement	ISO/IEC 12207 SLCP Reference	Target audience
Functional implementation coverage	How correct is the functional implementation?	Count the number of incorrectly implemented or missing functions and compare with the number of functions described in the requirement specifications.	X=1-A/B A= Number of incorrectly implemented or missing functions detected B= Number of functions described in requirement specifications	0 <= X <= 1 The closer to 1, the more correct.	absolute	X=count/count A=count B=count	Req spec Design Source code Review report	6.5 Validation 6.6 Joint review	Requirers Developers
FOOTNOTES									
1 Review by functional item.									
2 Input to the measurement process is the updated requirement specifications. Any changes identified during life cycle must be applied to the requirement specifications before using in measurement process.									
Functional specification stability (volatility)	How stable is the functional specification during the development life cycle?	Count the number of functions changed (added, modified, or deleted) during development life cycle phase, then compare with the number of functions described in the requirement specifications.	X=1-A/B A=Number of functions changed during development life cycle phases B=Number of functions described in requirement specifications	0 <= X <= 1 The closer to 1 the more stable.	absolute	A=Count B=Count X=Count/Count	Requirement specifications Review report	6.5 Validation 6.3 Quality Assurance 5.3 Qualification testing 6.8 Problem Resolution 5.4 Operation	Developers Maintainers

Table 8.1.2 Accuracy metrics

Internal accuracy metrics									
Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value	Metric scale type	Measure type	Input to measure-ment	ISO/IEC 12207 SLCP Reference	Target audience
Computational Accuracy	How completely have the accuracy requirements been implemented?	Count the number of functions that have implemented the accuracy requirements and compare with the number of functions with specific accuracy requirements.	X=A/B	0 <= X <= 1.	absolute	X=count/ count A=count B=count	Requirement specification Design Source code Review report	Verification Joint review	Requirers Developers
			A= Number of functions in which specific accuracy requirements had been implemented, as confirmed in evaluation B= Number of functions for which specific accuracy requirements need to be implemented	The closer to 1, the more complete.					
Precision	How complete was the implementation of specific levels of precision for the data items?	Count the number of data items that meet the requirements of specific levels of precision and compare to the total number of data items with specific level of precision requirements.	X=A/B A= Number of data items implemented with specific levels of precision, confirmed in evaluation B= Number of data items that require specific levels of precision	0 <= X <= 1. The closer to 1, the more complete.	absolute	X=count/ count A=count B=count	Requirement specification Design Source code Review report	Verification Joint review	Requirers Developers

Table 8.1.3 Interoperability metrics

Internal interoperability metrics									
Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value	Metric scale type	Measure type	Input to measure-ment	ISO/IEC 12207 SLCP Reference	Target audience
Data exchangeability (Data format based)	How correctly have the interface data formats been implemented?	Count the number of interface data formats that have been implemented correctly as in the specifications and compare to the number of data formats to be exchanged as in the specifications.	X=A/B	0 <= X <= 1.	absolute	X=count/ count A=count B=count	Req spec Design Source code Review report	Verification Joint review	Developers Requirers
			A=Number of interface data formats that have been implemented correctly as in the specifications B=Number of data formats to be exchanged as in the specifications	The closer to 1, the more correct.					
Interface consistency (protocol)	How correctly have the interface protocols been implemented?	Count the number of interface protocols that were implemented correctly as in the specifications and compare with the number of interface protocols to be implemented as in the specifications.	X=A/B A=Number of interface protocols implementing consistent format as in the specification confirmed in review B=Number of interface protocols to be implemented as in the specifications	0 <= X <= 1 The closer to 1, the more consistent.	absolute	X=count/ count A=count B=count	Req spec Design Source code Review report	Verification Joint review	Developers Requirers

Table 8.1.4 Security metrics

Internal security metrics									
Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value	Metric scale type	Measure type	Input to measurement	ISO/IEC 12207 SLCP Reference	Target audience
Access auditability	How auditable is access login?	Count the number of access types that are being logged correctly as in the specifications and compare with the number of access types that are required to be logged in the specifications.	X=A/B	0 <= X <= 1	Absolute	X=count/ count A=count B=count	Requirement specification	6.5	Requirers
			A= Number of access types that are being logged as in the specifications	The closer to 1, the more auditable.			Validation	Developers	
			B= Number of access types required to be logged in the specifications				Design		
				Source code			6.6		
						Review report	Joint review		
Access controllability	How controllable is access to the system?	Count the number of access controllability requirements implemented correctly as in the specifications and compare with the number of access controllability requirements in the specifications.	X=A/B	0 <= X <= 1	Absolute	X=count/ count A=count B=count	Requirement specification	6.5	Requirers
			A= Number of access controllability requirements implemented correctly as in the specifications	The closer to 1, the more controllable.			Validation	Developers	
			B= Number of access controllability requirements in the specifications				Design		
				Source code			6.6		
						Review report	Joint review		
Data corruption prevention	How complete is the implementation of data corruption prevention?	Count the number of implemented instances of data corruption prevention as specified and compare with the number of instances of operations/ access specified in requirements as capable of corrupting/ destroying data.	X=A/B	0 <= X <= 1	Absolute	X=count/ count A=count B=count	Requirement specification	6.5	Developers
			A= Number of implemented instances of data corruption prevention as specified confirmed in review	The closer to 1, the more complete.			Validation		
			B= Number of instances of operation/access identified in requirements as capable of corrupting/destroying data				Design	6.6	
							Source code	Joint review	
						Review report			
FOOTNOTE									
Consider security levels when using this metric.									
Data encryption	How complete is the implementation of data encryption?	Count the number of implemented instances of encryptable/decryptable data items as specified and compare with the number of instances of data items requiring data encryption/decryption facility as in specifications.	X=A/B	0 <= X <= 1	absolute	X=count/ count A=count B=count	Requirement specification	6.5	Validation
			A=Number of implemented instances of encryptable/decryptable data items as specified confirmed in review	The closer to 1, the more complete.					
			B= Number of data items requiring data encryption/decryption facility as in specifications				Design		
							Source code		
						Review report			
FOOTNOTE									
Data encryption: e.g., data in open database, data in public communication facility.									

Table 8.1.5 Functionality compliance metrics

Internal functionality compliance metrics									
Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value	Metric scale type	Measure type	Input to measurement	ISO/IEC 12207 SLCP Reference	Target audience
Functional compliance	How compliant is the functionality of the product to applicable regulations, standards and conventions?	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the specification.	X=A/B	0 <= X <= 1.	absolute	X=count/ count A=count B=count	Specification of compliance and related standards, conventions or regulations.	Verification	Requirers
			A= Number of correctly implemented items related to functionality compliance confirmed in evaluation B= Total number of compliance items	The closer to 1, the more compliant.			Design Source code Review report	Joint review Developers	
Intersystem standard compliance	How compliant are the interfaces to applicable regulations, standards and conventions?	Count the number of interfaces that meet required compliance and compare with the number of interfaces requiring compliance as in the specifications.	X=A/B	0 <= X <= 1.	absolute	X=count/ count A=count B=count	Req spec	Verification	Developers
			A= Number of correctly implemented interfaces as specified, confirmed in review B= Total number of interfaces requiring compliance	The closer to 1, the more compliant.			Design Source code Review report	Joint review Requirers	
FOOTNOTE									
All specified attributes of a standard must be checked.									

8.2 Reliability metrics

Internal reliability metrics are used for predicting if the software product in question will satisfy prescribed reliability needs, during the development of the software product.

8.2.1 Maturity metrics

Internal maturity metrics indicate a set of attributes for assessing the maturity of the software.

8.2.2 Fault tolerance metrics

Internal fault tolerance metrics indicate a set of attributes for assessing the software products capability in maintaining a desired performance level in case of operational faults or infringement of its specified interface.

8.2.3 Recoverability metrics

Internal recoverability metrics indicate a set of attributes for assessing the software product's capability to re-establish an adequate level of performance and recover the data directly affected in case of a failure.

8.2.4 Reliability compliance metrics

Internal compliance metrics relating to reliability indicate a set of attributes for assessing the capability of the software product to comply to such items as standards, conventions or regulations of the user organization in relation to reliability.