AS/NZS ISO/IEC 9126.2:2005
ISO/IEC TR 9126-2:2003
Reconfirmed 2016

AS/NZS ISO/IEC 9126.2:2005

Australian/New Zealand Standard™

Software engineering—Product quality

Part 2: External metrics

STANDARDS AUSTRALIA

STANDARDS NEW ZEALAND
AVA Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

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 Standards may also be withdrawn.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 05039.

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

STANDARDS AUSTRALIA/STANDARDS NEW ZEALAND

RECONFIRMATION

OF
AS/NZS ISO/IEC 9126.2:2005
Software engineering—Product quality
Part 2: External 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

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

NOTES

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

Australian/New Zealand Standard™

Software engineering—Product quality

Part 2: External metrics

First published as AS/NZS ISO/IEC 9126.2:2005.

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 6663 3

AVA Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

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 9126-2:2003, *Software engineering—Product qualities,* Part 2: *External metrics.*

The objective of this Standard is that 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.

This Standard is Part 2 of AS/NZS ISO/IEC 9126, *Software engineering—Product quality*, which is published in parts as follows:

Part 1: Quality model
Part 2: External metrics (this Standard)
Part 3: Internal metrics
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:

Reference to International Standard		*Australian/New Zealand Standard*	
ISO/IEC		AS/NZS	
9126	Software engineering—Product quality	9126	Software engineering—Product quality
9126-1	Part 1: Quality model	9126.1	Part 1: Quality model
9126-3	Part 3: Internal metrics	9126.3	Part 3: Internal metrics
9126-4	Part 4: Quality in use metrics	9126.4	Part 4: Quality in use metrics
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 acquires	14598.4	Part 4: Process for acquires
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

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

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.

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

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 5
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 6
8.2 Reliability metrics 14
8.2.1 Maturity metrics 14
8.2.2 Fault tolerance metrics 14
8.2.3 Recoverability metrics 14
8.2.4 Reliability compliance metrics 14
8.3 Usability metrics 25
8.3.1 Understandability metrics 25
8.3.2 Learnability metrics 25
8.3.3 Operability metrics 26
8.3.4 Attractiveness metrics 26
8.3.5 Usability compliance metrics 26
8.4 Efficiency metrics 41
8.4.1 Time behaviour metrics 41
8.4.2 Resource utilization metrics 41
8.4.3 Efficiency compliance metrics 41
8.5 Maintainability metrics 52
8.5.1 Analysability metrics 52
8.5.2 Changeability metrics 52
8.5.3 Stability metrics 52
8.5.4 Testability metrics 52
8.5.5 Maintainability compliance metrics 52
8.6 Portability metrics 60
8.6.1 Adaptability metrics 60

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

Page

8.6.2 Installability metrics 60
8.6.3 Co-existence metrics 60
8.6.4 Replaceability metrics 60
8.6.5 Portability compliance metrics 60
Annex A (informative) Considerations When Using Metrics 68
A.1 Interpretation of measures 68
A.1.1 Potential differences between test and operational contexts of use 68
A.1.2 Issues affecting validity of results 69
A.1.3 Balance of measurement resources 69
A.1.4 Correctness of specification 69
A.2 Validation of Metrics 69
A.2.1 Desirable Properties for Metrics 69
A.2.2 Demonstrating the Validity of Metrics 70
A.3 Use of metrics for estimation (judgement) and prediction (forecast) 71
A.3.1 Quality characteristics prediction by current data 71
A.3.2 Current quality characteristics estimation on current facts 71
A.4 Detecting deviations and anomalies in quality problem prone components 72
A.5 Displaying measurement results 72
Annex B (informative) Use of Quality in Use, External & Internal Metrics (Framework Example) 73
B.1 Introduction 73
B.2 Overview of Development and Quality Process 73
B.3 Quality Approach Steps 74
B.3.1 General 74
B.3.2 Step #1 Quality requirements identification 74
B.3.3 Step #2 Specification of the evaluation 75
B.3.4 Step #3 Design of the evaluation 77
B.3.5 Step #4 Execution of the evaluation 77
B.3.6 Step #5 Feedback to the organization 77
Annex C (informative) Detailed explanation of metric scale types and measurement types 78
C.1 Metric Scale Types 78
C.2 Measurement Types 79
C.2.1 Size Measure Type 79
C.2.2 Time measure type 82
C.2.2.0 General 82
C.2.3 Count measure type 83
Annex D (informative) Term(s) 85
D.1 Definitions 85
D.1.1 Quality 85
D.1.2 Software and user 85
D.1.3 Measurement 85

Table 8.1.1 Suitability metrics 7
Table 8.1.2 Accuracy metrics 9
Table 8.1.3 Interoperability metrics 10
Table 8.1.4 Security metrics 11

AVA Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

Page

Table 8.1.5 Functionality compliance metrics 13
Table 8.2.1 Maturity metrics 15
Table 8.2.2 Fault tolerance metrics 19
Table 8.2.3 Recoverability metrics 21
Table 8.2.4 Reliability compliance metrics 24
Table 8.3.1 Understandability metrics 27
Table 8.3.2 Learnability metrics 30
Table 8.3.3 Operability metrics a) Conforms with operational user expectations 32
Table 8.3.3 Operability metrics b) Controllable 33
Table 8.3.3 Operability metrics c) Suitable for the task operation 34
Table 8.3.3 Operability metrics d) Self descriptive (Guiding) 35
Table 8.3.3 Operability metrics e) Operational error tolerant (Human error free) 36
Table 8.3.3 Operability metrics f) Suitable for individualisation 37
Table 8.3.4 Attractiveness metrics 39
Table 8.3.5 Usability compliance metrics 40
Table 8.4.1 Time behaviour metrics a) Response time 42
Table 8.4.1 Time behaviour metrics b) Throughput 44
Table 8.4.1 Time behaviour metrics c) Turnaround time 45
Table 8.4.2 Resource utilisation metrics a) I/O devices resource utilisation 47
Table 8.4.2 Resource utilisation metrics b) Memory resource utilisation 48
Table 8.4.2 Resource utilisation metrics c) Transmission resource utilisation 49
Table 8.4.3 Efficiency compliance metrics 51
Table 8.5.1 Analysability metrics 53
Table 8.5.2 Changeability metrics 55
Table 8.5.3 Stability metrics 57
Table 8.5.4 Testability metrics 58
Table 8.5.5 Maintainability compliance metrics 59
Table 8.6.1 Adaptability metrics 61
Table 8.6.2 Installability metrics 63
Table 8.6.3 Co-existence metrics 65
Table 8.6.4 Replaceability metrics 66
Table 8.6.5 Portability compliance metrics 67
Table B.1 Quality Measurement Model 73
Table B.2 User Needs Characteristics & Weights 74
Table B.3 Quality measurement tables 75
Table B.4 Measurement plan 77

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

INTRODUCTION

This Technical Report provides external 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.

AVA Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

NOTES

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

AUSTRALIAN/NEW ZEALAND STANDARD

Software engineering — Product quality —

Part 2:
External metrics

1 Scope

This Technical Report defines external 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.

COPYRIGHT

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

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 9126-3[1)], *Software engineering — Product quality — Part 3: Internal metrics*

ISO/IEC 9126-4[1)], *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/IEC 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.

COPYRIGHT

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

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).

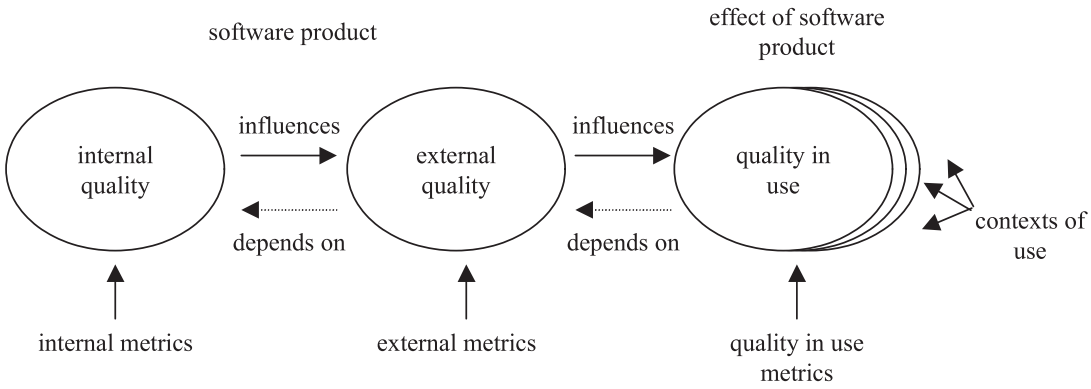

Figure 1 – Relationship between types of metrics

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.

COPYRIGHT

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

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.

COPYRIGHT

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

8.1 Functionality metrics

An external functionality metric should be able to measure an attribute such as the functional behaviour of a system containing the software. The behaviour of the system may be observed from the following perspectives:

a) Differences between the actual executed results and the quality requirements specification;

NOTE 1 The quality requirements specification for functionality is usually described as the functional requirements specification.

b) Functional inadequacy detected during real user operation which is not stated but is implied as a requirement in the specification.

NOTE 2 When implied operations or functions are detected, they should be reviewed, approved and stated in the specifications. Their extent to be fulfilled should be agreed.

8.1.1 Suitability metrics

An external suitability metric should be able to measure an attribute such as the occurrence of an unsatisfying function or the occurrence of an unsatisfying operation during testing and user operation of the system.

An unsatisfying function or operation may be:

a) Functions and operations that do not perform as specified in user manuals or requirement specification.

b) Functions and operations that do not provide a reasonable and acceptable outcome to achieve the intended specific objective of the user task.

8.1.2 Accuracy metrics

An external accuracy metric should be able to measure an attribute such as the frequency of users encountering the occurrence of inaccurate matters which includes:

a) Incorrect or imprecise result caused by inadequate data; for example, data with too few significant digits for accurate calculation;

b) Inconsistency between actual operation procedures and described ones in the operation manual;

c) Differences between the actual and reasonable expected results of tasks performed during operation.

8.1.3 Interoperability metrics

An external interoperability metric should be able to measure an attribute such as the number of functions or occurrences of less communicativeness involving data and commands, which are transferred easily between the software product and other systems, other software products, or equipment which are connected.

8.1.4 Security metrics

An external security metric should be able to measure an attribute such as the number of functions with, or occurrences of security problems, which are:

a) Failing to prevent leak of secure output information or data;

b) Failing to prevent loss of important data;

c) Failing to defend against illegal access or illegal operation.

COPYRIGHT

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

NOTE 1 It is recommended that penetration tests be performed to simulate attack, because such a security attack does not normally occur in the usual testing. Real security metrics may only be taken in “real life system environment”, that is “quality in use”.

NOTE 2 Security protection requirements vary widely from the case of a stand-alone-system to the case of a system connected to the Internet. The determination of the required functionality and the assurance of their effectiveness have been addressed extensively in related standards. The user of this standard should determine security functions using appropriate methods and standards in those cases where the impact of any damage caused is important or critical. In the other case the user may limit his scope to generally accepted “Information Technology (IT)” protection measures such as virus protection backup methods and access control.

8.1.5 Functionality compliance metrics

An external functionality compliance metric should be able to measure an attribute such as the number of functions with, or occurrences of compliance problems, which are the software product failing to adhere to standards, conventions, contracts or other regulatory requirements.

COPYRIGHT

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

COPYRIGHT

Table 8.1.1 Suitability metrics

External 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	Input to measure-ment	ISO/IEC 12207 SLCP Reference	Target audience
Functional adequacy	How adequate are the evaluated functions?	Number of functions that are suitable for performing the specified tasks comparing to the number of function evaluated.	X=1-A/B A= Number of functions in which problems are detected in evaluation B= Number of functions evaluated	0 <= X <= 1 The closer to 1.0, the more adequate.	Absolute	X= Count/ Count A= Count B= Count	Requirement specification (Req. Spec.) Evaluation report	6.5 Validation, 6.3 Quality Assurance, 5.3 Qualification testing	Developer, SQA
Functional implementation completeness	How complete is the implementation according to requirement specifications?	Do functional tests (black box test) of the system according to the requirement specifications. 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<=X<=1 The closer to 1.0 is the better.	Absolute	A= Count B= Count X= Count/ Count	Req. spec. Evaluation report	6.5 Validation, 6.3 Quality Assurance, 5.3 Qualification testing	Developer, SQA

FOOTNOTES

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

2 *This metric is suggested as experimental use.*

NOTE Any missing function cannot be examined by testing because it is not implemented. For detecting missing functions, it is suggested that each function stated in a requirement specification be tested one by one during functional testing. Such results become input to “Functional implementation completeness” metric. For detecting functions which are implemented but inadequate, it is suggested that each function be tested for multiple specified tasks. Such results become input to the “Functional adequacy” metric. Therefore, users of metrics are suggested to use both these metrics during functional testing.

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

Table 8.1.1 (*continued*)

External 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	Input to measure-ment	ISO/IEC 12207 SLCP Reference	Target audience
Functional implementation coverage	How correct is the functional implementation?	Do functional tests (black box test) of the system according to the requirement specifications. Count the number of incorrectly implemented or missing functions detected in evaluation and compare with the total number of functions described in the requirement specifications Count the number of functions that are complete versus the ones that are not.	X=1- A / B A= Number of incorrectly implemented or missing functions detected in evaluation B= Number of functions described in requirement specifications	0<=X<=1 The closer to 1.0 is the better.	Absolute	A= Count B= Count X= Count/ Count	Req. spec. Evaluation report	6.5 Validation, 6.3 Quality Assurance, 5.3 Qualification testing	Developer, SQA

FOOTNOTES

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

2 *This measure represents a binary gate checking of determining the presence of a feature.*

Functional specification stability (volatility)	How stable is the functional specification after entering operation?	Count the number of functions described in functional specifications that had to be changed after the system is put into operation and compare with the total number of functions described in the requirement specifications.	X = 1- A / B A= Number of functions changed after entering operation starting from entering operation B= Number of functions described in requirement specifications	0<=X<= 1 The closer to 1.0 is the better.	Absolute	A= Count B= Count X= Count/ Count	Req. spec. Evaluation report	6.8 Problem Resolution 5.4 Operation	Maintainer SQA

FOOTNOTE

This metric is suggested as experimental use.

COPYRIGHT

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

COPYRIGHT

Table 8.1.2 Accuracy metrics

External 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
Accuracy to expectation	Are differences between the actual and reasonable expected results acceptable?	Do input .vs. output test cases and compare the output to reasonable expected results. Count the number of cases encountered by the users with an unacceptable difference from reasonable expected results.	X=A / T A= Number of cases encountered by the users with a difference against to reasonable expected results beyond allowable T= Operation time	0<=X The closer to 0 is the better.	Ratio	A= Count T= Time X= Count/ Time	Req. spec. User operation manual Hearing to users Test report	6.5 Validation 6.3 Quality Assurance	Developer User
FOOTNOTE *Reasonable expected results might be identified in a requirement specification, a user operation manual, or users' expectations.*									
Computational Accuracy	How often do the end users encounter inaccurate results?	Record the number of inaccurate computations based on specifications.	X=A / T A= Number of inaccurate computations encountered by users T= Operation time	0<=X The closer to 0 is the better.	Ratio	A= Count T= Time X= Count/ Time	Req. spec. Test report	6.5 Validation 6.3 Quality Assurance	Developer User
Precision	How often do the end users encounter results with inadequate precision ?	Record the number of results with inadequate precision.	X=A / T A= Number of results encountered by the users with level of precision different from required T= Operation time	0<=X The closer to 0 is the better.	Ratio	A= Count T= Time X= Count/ Time	Req. spec. Test report	6.5 Validation 6.3 Quality Assurance	Developer User

NOTE Data elements for computation of external metrics are designed to use externally accessible information, because it is helpful for end users, operators, maintainers or acquirers to use external metrics. Therefore, the time basis metric often appears in external metrics and is different from internal ones.

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

Table 8.1.3 Interoperability metrics

External 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 exchange interface functions for specified data transfer been implemented?	Test each downstream interface function output record format of the system according to the data fields specifications. Count the number of data formats that are approved to be exchanged with other software or system during testing on data exchanges in comparing with the total number.	X= A / B A= Number of data formats which are approved to be exchanged successfully with other software or system during testing on data exchanges B= Total number of data formats to be exchanged	0<=X<= 1 The closer to 1.0 is the better.	Absolute	A= Count B= Count X= Count/ Count	Req. spec. (User manual) Test report	6.5 Validation	Developer
FOOTNOTE *It is recommended to test specified data transaction.*									
Data exchangeability (User's success attempt based)	How often does the end user fail to exchange data between target software and other software? How often are the data transfers between target software and other software successful? Can user usually succeed in exchanging data?	Count the number of cases that interface functions were used and failed.	a) X= 1 - A / B A= Number of cases in which user failed to exchange data with other software or systems B= Number of cases in which user attempted to exchange data b) Y= A / T T= Period of operation time	0<=X<= 1 The closer to 1.0 is the better. 0<=Y The closer to 0, is the better.	a) Absolute b) Ratio	A= Count B= Count X= Count/ Count Y= Count/ Time T= Time	Req. spec. (User manual) Test report	5.4 Operation	Maintainer

COPYRIGHT

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

COPYRIGHT

Table 8.1.4 Security metrics

External 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 measure-ment	ISO/IEC 12207 SLCP Reference	Target audience
Access auditability	How complete is the audit trail concerning the user access to the system and data?	Evaluate the amount of accesses that the system recorded in the access history database.	X= A / B A= Number of "user accesses to the system and data" recorded in the access history database B= Number of " user accesses to the system and data" done during evaluation	0<=X<=1 The closer to 1.0 is the better.	Absolute	A= Count B= Count X= Count/ Count	Test spec. Test report	6.5 Validation	Developer
FOOTNOTES *1 Accesses to data may be measured only with testing activities.* *2 This metric is suggested as an experimental use.* *3 It is recommended that penetration tests be performed to simulate attacks, because such security attacks do not normally occur in the usual testing. Real security metrics may only be taken in "real life system environment", that is "quality in use".* *4 "User access to the system and data" record may include "virus detection record" for virus protection. The aim of the concept of computer virus protection is to create suitable safeguards with which the occurrence of computer viruses in systems can be prevented or detected as early as possible.*									
Access controllability	How controllable is access to the system?	Count number of detected illegal operations with comparing to number of illegal operations as in the specification.	X= A / B A= Number of detected different types of illegal operations B= Number of types of illegal operations as in the specification	0<=X<=1 The closer to 1.0 is the better.	Absolute	A= Count B= Count X= Count/ Count	Test spec. Test report Operation report	6.5 Validation 6.3 Quality Assurance	Developer
FOOTNOTES *1 If it is necessary to complement detection of unexpected illegal operations additional intensive abnormal operation testing should be conducted.* *2 It is recommended that penetration tests be performed to simulate attack, because such security attacks do not normally occur in the usual testing. Real security metrics may only be taken in "real life system environment", that is "quality in use".* *3 Functions prevent unauthorized persons from creating, deleting or modifying programs or information. Therefore, it is suggested to include such illegal operation types in test cases.*									

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

Table 8.1.4 (*continued*)

External 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 measure-ment	ISO/IEC 12207 SLCP Reference	Target audience
Data corruption prevention	What is the frequency of data corruption events?	Count the occurrences of major and minor data corruption events.	a) X= 1 – A / N A= Number of times that a major data corruption event occurred N= Number of test cases tried to cause data corruption event	0<=X<= 1 The closer to 1.0 is the better.	a) Absolute	A= Count B= Count N= Count X= Count/ Count	Test spec. Test report Operation report	6.5 Validation 5.3 Qualification testing 5.4 Operation	Maintainer Developer
			b) Y= 1- B / N B= Number of times that a minor data corruption event occurred	0<=Y<= 1 The closer to 1.0 is the better.	b) Absolute	Y= Count/ Count			
			c) Z= A / T or B / T T= period of operation time (during operation testing)	0<=Z The closer to 0, is the better.	c) Ratio	T= Time Z= Count/ Time			

FOOTNOTES

1 *Intensive abnormal operation testing is needed to obtain minor and major data corruption events.*

2 *It is recommended to grade the impact of data corruption events such as the following examples:*
Major (fatal) data corruption event:
- reproduction and recovery impossible;
- second affection distribution too wide;
- importance of data itself.
Minor data corruption event:
- reproduction or recovery possible and
- no second affection distribution;
- importance of data itself.

3 *Data elements for computation of external metrics are designed to use externally accessible information, because it is helpful for end users, operators, maintainers or acquirers to use external metrics. Therefore, counting events and times used here are different from corresponding internal metric.*

4 *It is recommended that penetration tests be performed to simulate attack, because such security attacks do not normally occur in the usual testing.*
Real security metrics may only be taken in "real life system environment", that is "quality in use"

5 *This metric is suggested as an experimental use.*

6 *Data backup is one of the effective ways to prevent data corruption. The creation of back up ensures that necessary data can be restored quickly in the event that parts of the operative data are lost. However, data back up is regarded as a part of the composition of the reliability metrics in this report.*

7 *It is suggested that this metric be used experimentally.*

COPYRIGHT

AVA Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

COPYRIGHT

Table 8.1.5 Functionality compliance metrics

External 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 measure-ment	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 in the specification. Design test cases in accordance with compliance items. Conduct functional testing for these test cases. Count the number of compliance items that have been satisfied.	X = 1 - A / B A= Number of functionality compliance items specified that have not been implemented during testing B= Total number of functionality compliance items specified	0<= X <=1 The closer to 1.0 is the better.	Absolute	A= Count B= Count X= Count/ Count	Product description (User manual or Specification) of compliance and related standards, conventions or regulations Test specification and report	5.3 Qualification testing 6.5 Validation	Supplier User

FOOTNOTES

1 *It may be useful to collect several measured values along time, to analyse the trend of increasingly satisfied compliance items and to determine whether they are fully satisfied or not.*

2 *It is suggested to count number of failures, because problem detection is an objective of effective testing and also suitable for counting and recording.*

Interface 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 A= Number of correctly implemented interfaces as specified B= Total number of interfaces requiring compliance	0<=X<= 1 The closer to 1.0 is the better.	Absolute	A= Count B= Count X= Count/ Count	Product description of compliance and related standards, conventions or regulations Test specification and report	6.5 Validation	Developer

FOOTNOTE

All specified attributes of a standard must be tested.

Documents
Document Sharing Hub

8.2 Reliability metrics

An external reliability metric should be able to measure attributes related to the behaviours of the system of which the software is a part during execution testing to indicate the extent of reliability of the software in that system during operation. Systems and software are not distinguished from each other in most cases.

8.2.1 Maturity metrics

An external maturity metric should be able to measure such attributes as the software freedom of failures caused by faults existing in the software itself.

8.2.2 Fault tolerance metrics

An external fault tolerance metric should be related to the software capability of maintaining a specified performance level in cases of operation faults or infringement of its specified interface.

8.2.3 Recoverability metrics

An external recoverability metric should be able to measure such attributes as the software with system being able to re-establish its adequate level of performance and recover the data directly affected in the case of a failure.

8.2.4 Reliability compliance metrics

An external reliability compliance metric should be able to measure an attribute such as the number of functions with, or occurrences of compliance problems, in which the software product fails to adhere to standards, conventions or regulations relating to reliability.

COPYRIGHT

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

COPYRIGHT

Table 8.2.1 Maturity metrics

External maturity 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
Estimated latent fault density	How many problems still exist that may emerge as future faults?	Count the number of faults detected during a defined trial period and predict potential number of future faults using a reliability growth estimation model.	X= {ABS(A1 - A2)} / B (X: Estimated residuary latent fault density) ABS()= Absolute Value A1 = total number of predicted latent faults in a software product A2 = total number of actually detected faults B= product size	0<=X It depends on stage of testing. At the later stages, smaller is better.	Absolute	A1= Count A2= Count B= Size X= Count/ Size	Test report Operation report Problem report	5.3 Integration 5.3 Qualification testing 5.4 Operation 6.5 Validation 6.3 Quality Assurance	Developer Tester SQA User

FOOTNOTES

1 *When total number of actually detected faults becomes larger than total number of predicted latent faults, it is recommended to predict again and estimate more larger number. Estimated larger numbers are intended to predict reasonable latent failures, but not to make the product look better.*

2 *It is recommended to use several reliability growth estimation models and choose the most suitable one and repeat prediction with monitoring detected faults.*

3 *It may be helpful to predict upper and lower number of latent faults.*

4 *It is necessary to convert this value (X) to the <0,1> interval if making summarisation of characteristics.*

Failure density against test cases	How many failures were detected during defined trial period?	Count the number of detected failures and performed test cases.	X= A1 / A2 A1 = number of detected failures A2 = number of performed test cases	0<=X It depends on stage of testing. At the later stages, smaller is better.	Absolute	A1= Count A2= Count B = Size X= Count/ Count	Test report Operation report Problem report	5.3 Integration 5.3 Qualification testing 5.4 Operation 6.3 Quality Assurance	Developer Tester SQA

FOOTNOTES

1 *The larger is the better, in early stage of testing. On the contrary, the smaller is the better, in later stage of testing or operation. It is recommended to monitor the trend of this measure along with the time.*

2 *This metric depends on adequacy of test cases so highly that they should be designed to include appropriate cases: e.g., normal, exceptional and abnormal cases.*

3 *It is necessary to convert this value (X) to the <0,1> interval if making summarisation of characteristics.*

AVA Documents Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

Table 8.2.1 (*continued*)

External maturity 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
Failure resolution	How many failure conditions are resolved?	Count the number of failures that did not reoccur during defined trial period under the similar conditions. Maintain a problem resolution report describing status of all the failures.	X= A1 / A2 A1 = number of resolved failures A2 = total number of actually detected failures	0<=X<= 1 The closer to 1.0 is better as more failures are resolved.	a) Absolute	A1= Count A2= Count A3 = Count X= Count/ Count	Test report Operation (test) report	5.3 Integration 5.3 Qualification testing 5.4 Operation	User SQA Maintainer
FOOTNOTES *1 It is recommended to monitor the trend when using this measure.* *2 Total number of predicted latent failures might be estimated using reliability growth models adjusted with actual historical data relating to similar software product. In such a case, the number of actual and predicted failures can be comparable and the number of residual unresolved failures can be measurable.*									
Fault density	How many faults were detected during defined trial period?	Count the number of detected faults and compute density.	X= A / B A = number of detected faults B = product size	0<=X It depends on stage of testing. At the later stages, smaller is better.	Absolute	A= Count B = Size X= Count/ Size	Test report Operation report Problem report	5.3 Integration 5.3 Qualification testing 5.4 Operation 6.3 Quality Assurance	Developer Tester SQA
FOOTNOTES *1 The larger is the better, in early stage of testing. On the contrary, the smaller is the better, in later stage of testing or operation. It is recommended to monitor the trend of this measure along with the time.* *2 The number of detected faults divided by the number of test cases indicates effectiveness of test cases.* *3 It is necessary to convert this value (X) to the <0,1> interval if making summarisation of characteristics.* *4 When counting faults, pay attention to the followings:* *- Possibility of duplication, because multiple reports may contain the same faults as other report;* *- Possibility of others than faults, because users or testers may not figure out whether their problems are operation error, environmental error or software failure.*									

COPYRIGHT

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

COPYRIGHT

Table 8.2.1 (*continued*)

External maturity 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
Fault removal	How many faults have been corrected?	Count the number of faults removed during testing and compare with the total number of faults detected and total number of faults predicted.	a) X= A1 / A2 A1 = number of corrected faults A2 = total number of actually detected faults b) Y= A1 / A3 A3 = total number of predicted latent faults in the software product	0<=X<= 1 The closer to 1.0 is better as fewer faults remain. 0<=Y The closer to 1.0 is better as fewer faults remain.	a) Absolute b) Absolute	A1= Count A2= Count A3= Count X= Count/ Count Y= Count/ Count	Test report Organization database	5.3 Integration 5.3 Qualification testing 6.5 Validation 6.3 Quality Assurance	Developer SQA Maintainer

FOOTNOTES

1 *It is recommended to monitor the trend during a defined period of time.*

2 *Total number of predicted latent faults may be estimated using reliability growth models adjusted with actual historical data relating to similar software product.*

3 *It is recommended to monitor the estimated faults resolution ratio Y, so that if Y > 1, investigate the reason whether it is because more faults have been detected early or because software product contains an unusual number of faults.*
Otherwise, when Y < 1, investigate whether it is because there are less than the usual number of defects in the software products or because the testing was not adequate to detect all possible faults.

4 *It is necessary to convert this value (Y) to the <0,1> interval if making summarisation of characteristics.*

5 *When counting faults, pay attention to the possibility of duplication, because multiple reports may contain the same faults as other report.*

Mean time between failures (MTBF)	How frequently does the software fail in operation?	Count the number of failures occurred during a defined period of operation and compute the average interval between the failures.	a) X = T1 / A b) Y = T2 / A T1 = operation time T2 = sum of time intervals between consecutive failure occurrences A = total number of actually detected failures (Failures occurred during observed operation time)	0<X,Y The longer is the better as longer time can be expected between failures.	a) Ratio b) Ratio	A = Count T1 = Time T2 = Time X =Time / Count Y =Time/ Count	Test report Operation (test) report	5.3 Integration 5.3 Qualification testing 5.4 Operation testing 5.4 Operation	Maintainer User

FOOTNOTES

1 *The following investigation may be helpful: - distribution of time interval between failure occurrences;*
- changes of mean time along with interval operation time period;
- distribution indicating which function has frequent failure occurrences and operation because of function and use dependency.

2 *Failure rate or hazard rate calculation may be alternatively used.*

3 *It is necessary to convert this value (X,Y) to the <0,1> interval if making summarisation of characteristics.*

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

Table 8.2.1 (*continued*)

External maturity 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
Test coverage (Specified operation scenario testing coverage)	How much of required test cases have been executed during testing?	Count the number of test cases performed during testing and compare the number of test cases required to obtain adequate test coverage.	X= A / B A= Number of actually performed test cases representing operation scenario during testing B= Number of test cases to be performed to cover requirements	0<=X<=1 The closer to 1.0 is the better test coverage.	Absolute	A= Count B= Count X= Count/ Count	Req. spec., Test spec. or User manual Test report Operation report	5.3 Qualification testing 6.5 Validation 6.3 Quality Assurance	Developer Tester SQA
FOOTNOTE *Test cases may be normalised by software size, that is: test density coverage Y= A / C, where C= Size of product to be tested. The larger Y is the better. Size may be functional size that user can measure.*									
Test maturity	Is the product well tested? ***COMMENT(S)*** *This is to predict the success rate the product will achieve in future testing.*	Count the number of passed test cases which have been actually executed and compare it to the total number of test cases to be performed as per requirements.	X= A / B A= Number of passed test cases during testing or operation B= Number of test cases to be performed to cover requirements	0<=X<=1 The closer to 1.0 is the better.	Absolute	A= Count B= Count X= Count/ Count	Req. spec., Test spec., or User manual Test report Operation report	5.3 Qualification testing 6.3 Quality Assurance	Developer Tester SQA
FOOTNOTES 1 *It is recommended to perform stress testing using live historical data especially from peak periods. It is also recommended to ensure that the following test types are executed and passed successfully:* *- User operation scenario;* *- Peak stress;* *- Overloaded data input.* 2 *Passed test cases may be normalised by software size, that is: passed test case density Y= A / C, where C= Size of product to be tested. The larger Y is better. Size may be functional size that user can measure.*									

COPYRIGHT

Documents
Document Sharing Hub

Full document details: AS NZS ISO IEC 9126.2-2005 (2016)

COPYRIGHT

AVA Documents
Document Sharing Hub

Table 8.2.2 Fault tolerance metrics

External fault tolerance 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
Breakdown avoidance	How often the software product causes the break down of the total production environment?	Count the number of breakdowns occurrence with respect to number of failures. If it is under operation, analyse log of user operation history.	X= 1- A / B A= Number of breakdowns B= Number of failures	0<=X<= 1 The closer to 1.0 is the better.	Absolute	A =Count B =Count X =Count/ Count	Test report Operation report	5.3 Integration 5.3 Qualification testing 5.4 Operation	User Maintainer

FOOTNOTES

1 *The breakdown means the execution of any user tasks is suspended until system is restarted, or its control is lost until system is forced to be shut down.*

2 *When none or few failures are observed, time between breakdowns may be more suitable.*

Failure avoidance	How many fault patterns were brought under control to avoid critical and serious failures?	Count the number of avoided fault patterns and compare it to the number of fault patterns to be considered	X=A / B A= Number of avoided critical and serious failure occurrences against test cases of fault pattern B= Number of executed test cases of fault pattern (almost causing failure) during testing	0<=X<= 1 The closer to 1.0 is better, as the user can more often avoid critical or serious failure.	Absolute	A= Count B= Count X= Count/ Count	Test report Operation report	5.3 Integration 5.3 Qualification testing 5.4 Operation 6.5 Validation	User Maintainer

FOOTNOTES

1 *It is recommended to categorise failure avoidance levels which is the extent of mitigating impact of faults, for example:*
-Critical: entire system stops / or serious database destruction;
-Serious: important functions become inoperable and no alternative way of operating (workaround);
-Average: most functions are still available, but limited performance occurs with limited or alternate operation (workaround);
-Small: a few functions experience limited performance with limited operation;
-None: impact does not reach end user.

2 *Failure avoidance levels may be based on a risk matrix composed by severity of consequence and frequency of occurrence provided by ISO/IEC 15026 System and software integrity.*

3 *Fault pattern examples*
- out of range data
- deadlock
Fault tree analysis technique may be used to detect fault patterns.

4 *Test cases can include the human incorrect operation.*