TECHNICAL REPORT

ISO/IEC TR 9126-2

First edition
2003-07-01

Software engineering — Product quality —

Part 2: External metrics

Génie du logiciel — Qualité des produits — Partie 2: Métrologie externe

Reference number
ISO/IEC TR 9126-2:2003(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, the joint technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

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

ISO/IEC TR 9126-2:2003, which is a Technical Report of type 2, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and system engineering*.

This document is being issued in the Technical Report (type 2) series of publications (according to the Procedures for the technical work of ISO/IEC JTC 1) as a "prospective standard for provisional application" in the field of external metrics for quantitatively measuring external software because there is an urgent need for guidance on how standards in this field should be used to meet an identified need.

This document is not to be regarded as an "International Standard". It is proposed for provisional application so that information and experience of its use in practice may be gathered. Comments on the content of this document should be sent to the ISO Central Secretariat.

A review of this Technical Report (type 2) will be carried out not later than three years after its publication with the options of: extension for another three years; conversion into an International Standard; or withdrawal.

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

— *Part 1: Quality model*

— *Part 2: External metrics*

— *Part 3: Internal metrics*

— *Part 4: Quality in use metrics*

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.

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.

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.

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).

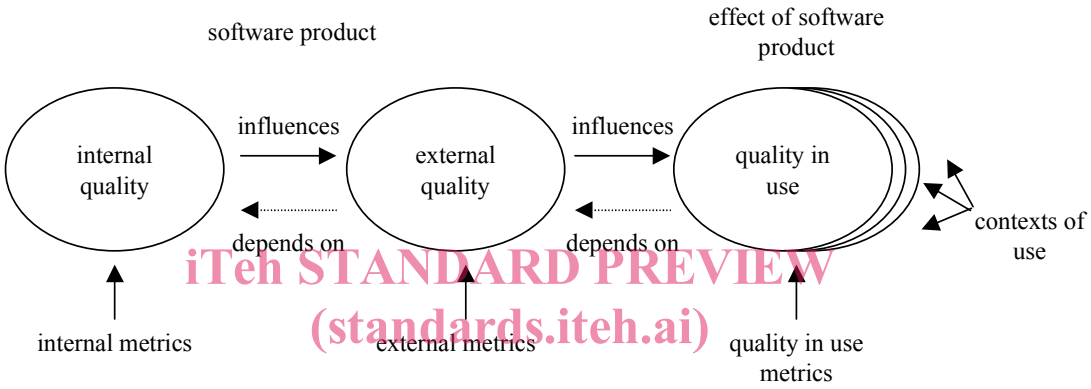

Figure 1 – Relationship between types of metrics

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

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.

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.

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.