ISO copyright Office
Case postale 56
CH-1211 Geneva 20
Fax +41 22 749 09 47
E-mail copyright@iso.org

6/N-456

ISO/IEC JTC1/SC7/WG6

EVALUATION AND METRICS

TITLE	ISO/IEC 9126-2 : Information Technology – Software product quality –Part 2 : External Metrics
DATE:	Nov. 17, 1999
SOURCE:	ISO/IEC JTC1/SC7/WG6
WORK ITEM:	Project 7-13-01-02
STATUS:	Version 5.8 – reflects ISO/IEC JTC1/SC7/WG6 Crituba, Brazil meeting in May, 1999 and comments of 2[nd] PDTR Ballot
REFERENCE:	6N/443(7N2073), 6/N 425,WG6-ROMA12R, 7N1809, 6/N 403R(7N 1767) ,7N1669, 6/N 363 (7N1563)
DOCUMENT TYPE:	Preliminal Draft Technical Report
ACTION:	For preliminary work for DTR Ballot.
EDITOR:	Atsushi Yamada System Integration Technology Center, Information and Industrial Systems & Services Company, Toshiba Corporation 3-22, Katamachi, Fuchu-shi, Tokyo 183-8512 , Japan TEL: +81-42-340-6399 , FAX: +81-42-340-6013 E-mail: atsu.yamada@toshiba.co.jp
CO-EDITOR:	Vipula GODAMUNNE IBM Global Services Australia TEL: +61-2-9353-3727 , FAX: - E-mail: vipula@au1.ibm.com
REVIEWERS	Ms. Antonia Jeanrenaud (Italy), Mr. Dave Hufton (Singapore), Dr. Jili Vanicek (Czech Republic), Mr. Vipula Godamunne (Australia), Dr. Alain Abran (Canada), Ms. Carole Nadeau (Canada: Previous Co-Editor), Mr. Alain April (Canada: Previous Co-Editor) Dr. Chris Vermaak(Sauth Africa : Previous Co-Editor)

REVISION HISTORY

Document #	Version	Date	Description
6/N 456	5.8	Nov. 99	Prelimianal work for Draft techinical report for DTR ballot reflecting 2nd PDTR ballot comments and Critiba, Brazil meeting in May 1999. It is prepared for Knazawa, Japan meeting in Nov. 99.
6/N 443	5.7	Jan. 99	Proposed draft techinical report for 2nd PDTR ballot reflecting Sydney meeting discussion
6/N	5.6	Nov.98	Version 5.6 – revised working draft after PDTR registration (for Sydney, Australia meeting Nov. 23, 1998)
6/N 425	5.5	May. 98	Working version for South Africa meeting discussion, which is under revising to resolve both of ballot comments before Roma meeting and additional comments after Roma.
Roma 12R	5.4	Nov.97	Working version reflecting discussion and work to coordinate 9126 part2 and 3 as a meeting work product.
Roma 12	-	Nov. 97	Working version reflecting discussion and work to coordinate 9126 part2 and 3 as a intermediate meeting work product.
7N 1809	-	Nov. 97	PDTR letter ballot summary - approved with comments
7N 1767 6/N 403R	5.3.2	Jun. 97	Working version after the review in California June 1997 and circulated in SC7 for PDTR Ballot
403R	5.3.2, 5.3.1 5.3	Jun. 97	Working version after the review in California June 1997 (3rd Walnut Creek meeting version proposing more widely resolutions to the ballot comments)
Walnut Creek 14R	5.2	Jun. 97	Working version for review in California June 1997 (2nd Walnut Creek meeting version proposing more widely resolutions to the ballot comments)
Walnut Creek 14	5.1	Jun. 97	Working version for review in California June 1997 (1st Walnut Creek meeting version proposing more widely resolutions to the ballot comments)
7N 1669	-	Feb. 97	WD & PDTR Registration letter ballot summary - approved with comments
	5.0	Jan. 97	Working version for review in California June 1997
Curitiba	4.9	Nov. 96	Working document for review at Curitiba meeting
6/N 363	4.8	July 96	Working document for PDTR after Prague meeting May 1996 and circulated in SC7 for WD & PDTR Registration Ballot
6/N 320		May. 96	Working document for review at Prague meeting May 1996
		May. 95	Working document for review at Brisbane meeting May 1995
		May. 94	Working document for review at Ottawa meeting May 1994

Reference number of working document: **ISO/IEC JTC1 /SC7 /WG6 6N456**

Date: 1999-11-17

Reference number of document: **ISO/IEC DTR 9126-2**

Committee identification: ISO/IEC JTC1 /SC 7/WG 6

Secretariat: Canada and Japan

Information Technology –

Software Product Quality – part2: External Metrics

Titre — Titre — Partie n: Titre

Document type: International technical report
Document subtype: if applicable
Document stage: (40) Enquiry
Document language: E

ISO Basic template Version 3.0 1997-02-03

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for world-wide 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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. 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.

International Technical Report ISO/IEC 9126-2 was prepared by the Joint Technical Committee ISO/IEC JTC1, Information Technology, Subcommittee SC7, Software Engineering.

Introduction

This international technical report (ITR) provides external metrics for measuring attributes of six quality characteristics defined in ISO/IEC 9126-1. This report also provides data elements, which are commonly used for composing metrics. The metrics and data elements listed in this ITR are considered to be a basic set. Developers, evaluators, quality managers and acquirers may select metrics and data elements from this technical report for defining requirements, evaluating software products, measuring quality aspects and other purposes. They may also use metrics and data elements, which are not included here. This report is applicable to any kind of software product, although each of the metrics is not always applicable to every kind of software.

ISO/IEC 9126-1 defines terms for global characteristics and how these characteristics are decomposed into subcharacteristics. ISO/IEC 9126-1, however, does not describe how to state requirements with respect to subcharacteristics, or how, for a given product, any of these subcharacteristics could be measured. These two objectives cannot be achieved by further decomposing the subcharacteristics recursively in the same way. Something new is required. To attempt to partially fill this gap between the subcharacteristic concept (ISO/IEC 9126-1) and a measurable characteristic, technical reports are presented: 9126-2 on external metrics, 9126-3 on internal metrics and 9126-4 quality-in-use metrics.

Software Product Quality – part2 : External Metrics

1 Scope

This international technical report (ITR) presents external metrics for quantitatively measuring software quality in terms of characteristics and subcharacteristics defined in ISO/IEC 9126-1 and explains methods for determining characteristics values from attributes values.

This ITR 9126-2 External Metrics presents:

- a general intent of quality metrics (Clause 6)

- which measurable attributes contribute to which software quality characteristics (including subcharacteristics) so that they can serve as metrics for these characteristics

- the metrics' desirable properties

a quality approach experimentation example

This report contains the terminology related to the metrics measurements, the usage of metrics in the life cycle process, and the introductory suggested sets of external and quality-in-use metrics for each software quality characteristic and subcharacteristic. This report provides a guide to the user of metrics for planning evaluations, selecting metrics, designing metrics, applying metrics, and interpreting measurement data.

NOTE : Out of scope

This ITR 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 ITR is allowed to be applied to any kind of software for any application, because this ITR does not include any requirements for specific application domain. Therefore, users of this ITR can select or develop and apply the suitable metrics and measures which may be from this ITR or any other references for their individual application domain. For example, the specific measurement of quality characteristics such as safety, security and human factors when these are mission-critical. For critical applications, it is highly recommended to use other specific standards that give precise guidance on how to achieve and measure the required critical quality characteristics such as safety, security, and human factors (for example, these may be found in IS or ITR provided by IEC 65 and JTC1/SC27).

Users of this ITR include, as an example:

1. Acquirer (an organization that acquires or procures a system, software product or software service from a supplier) ;

2. Evaluator (an 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 an user) ;

3. Developer (an organization that performs development activities, including requirements analysis, design, and testing through acceptance during the software life cycle process) ;

4. Maintainer (an organization that performs maintenance activities) ;

5. Supplier (an 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;

6. User (an individual or organization that uses the software product to perform a specific function) when evaluating quality of software product at acceptance test;

7. Quality managers (an organization that perform an systematic examination of the software product or software services) when evaluating software quality at qualification test.

2 Conformance

Since this is an ITR, there are no conformance requirements. Organizations and/or users (acquirers, developers, etc.) should stipulate which clauses of this ITR are needed, as stated in the Introduction.

3 Reference(s)

ISO 8402 : 1994, Quality management and quality assurance – Quality Vocaburary

ISO/IEC 9126 : 1991, Information technology - Software product evaluation – Quality characteristics and guidelines for their use

ISO/IEC 9126-1(new) : Information Technology - Software product quality - Part 1: Quality model

ISO/IEC 9126-2(new) : Information Technology - Software product quality - Part 2: External metrics

ISO/IEC 9126-4(new) : Information Technology - Software product quality - Part 4: Quality in use metrics

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

SO/IEC 14598-2: 1999, Information Technology - Software product evaluation - Part 2: Planning and management

ISO/IEC 14598-3: 1999, Information Technology - Software product evaluation - Part 3: Process for developers

ISO/IEC 14598-4: 1999, Information Technology - Software 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(new) : Information Technology - Software product evaluation - Part 6: Documentation of evaluation modules

ISO/IEC 12207 : 1995, Information technology – Software life-cycle processes.

ISO/IEC 14143 : 1998, Functional size measurement

ISO 2382-20:1990, Information technology, Vocabulary

4 Term(s) and definition(s)

For the purposes of this ISO/IEC 9126-2, the definitions contained in ISO/IEC 14598-1 and ISO/IEC 9126-1apply.

5 Symbols (and abbreviated terms)

The following symbols and abbreviations are used in this ITR:

SQA Software Quality Assurance (Group)

6 General intent of the quality metrics

These reports (9126-2, 9126-3 and 9126-4) provides a suggested set of quality metrics (external, quality in use and internal metrics) to be used with the 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 ITRs, the user should specify how the metrics relate to the 9126-1 quality model or any other substitute quality model that is being used.

The user of these ITRs may select the quality characteristics and subcharacteristics to be evaluated, from ISO/IEC 9126-1; identify the appropriate direct and indirect measures to be applied, identify the relevant metrics and then interpret the measurement result in a objective manner. The user of these ITRs also may select product quality evaluation processes during software life-cycle from 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.

The metrics desirable properties (clause 6.3) should be applied to the new or modified metric.

6.1 Concept descriptions

This sub clause will reinforce the ISO/IEC 9126-1 concepts of internal, external and quality in use metrics before explaining how to use these quality metrics.

The internal metrics may be applied to a non-executable software product during it's 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 detect quality issues and take corrective actions during the early stages of the development life cycle process.

The external metrics may be used to measure the quality of the software product by measuring the behavior of the system of which it is a part. The external metrics should only be used during the testing stages of the life cycle process and during any operational stages. This is achieved by executing the software product in the system environment that it is intended for. (It can be performed in in-house testing site or independent testing laboratory over the system environment site that it is intended for.)

The quality in use metrics measure the extent to which 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 real life system environment.

What bind these three types of metrics (internal, external and quality in use) in an objective manner during an evaluation process are the quality criteria to be used for determining the quality of the software product. The metrics should be applied in a manner that supplements each other during the evaluation.

The process driving mentioned in the above paragraphs could be implemented in the following way. When the software quality requirements are defined, the software quality characteristics or subcharacteristics, which represent the quality requirements, are listed. Then, the appropriate external metrics and acceptable ranges are specified to quantify the quality criteria, which validate that the software meets the user needs. The internal quality attributes of the software are then defined and specified to plan to achieve the required external quality characteristics finally and to build them into the intermediate product during development. Appropriate internal metrics and acceptable range are specified to quantify the internal quality characteristics so that they can be used for verifying that the intermediate software meets the internal quality specifications during the development. Quality in use metrics measure the extent to which a product meets the needs of specified users to achieve specified goals with effectiveness, productivity, safety and satisfaction, when the software is consequently operated as a part of the system in the individual user's environment. The relationship with other quality characteristics depends on the type of the user.

It is recommended the use of internal metrics having a relation 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, which provides a strong relationship between internal metrics and external metrics. Therefore, a hypothetical model containing ambiguity may be designed and the extent of the relationship may be modeled statistically during the use of metrics.

In relation to these definitions the next sub clause will present the interpretation of these new concepts.

6.1.1 External metrics

External metrics should be designed to:

1.represent software product quality which includes software quality characteristics and subcharacteristics defined in ISO/IEC 9126-1, during testing or operation;

2.validate that the software satisfies external quality requirements;

3.predict the actual quality in use;

4.describe the extent to which the software product keeps on satisfying user's stated or implied needs during the actual operation by user.

External metrics should be designed to employ the following measurements:

1.Measurements of software behavior in testing and operating, and in cooperation with other software, hardware, or system;

2.Measurements of user behavior.

NOTES:

1.Measurements of software behavior include measurements of incidents which are threats to human life and health, environmental natural resources, data destruction, inconsistency or misleading of information, broken security, degrade of service, advantages or profits in a market, gain or loss of economy etc.

2.The external metrics may be used to predict and to indicate the density of potential faults remaining in the software.

3.Internal measurements may be used to calculate external measures. For example, the number of program steps, which is an internal measure, may be used to normalize an external measure, such as failure occurrences against size of software, that is, failure density during testing.

6.1.2 Internal metrics

Internal metrics provide users, evaluators, testers, developers, quality managers, or managers with benefits whereby they are able to evaluate software quality of intermediate and final products during the time period before the software product is executable.

Internal metrics should be designed to:

1.represent software quality of intermediate and final products for characteristics which includes software quality characteristics and subcharacteristics defined in ISO/IEC 9126-1, during software product is not executable;

2.guide to planning and implementation of designs, programs or processes improvement, which affect intermediate and final software products;

3.verify that the intermediate and final software products satisfy internal quality requirements which are quality improvement plans for designs, programs or processes;

4.predict the external metrics or quality.

Internal metrics should be designed to employ following measurement:

Measurements of static software attributes which have appeared in the text of source code, in a graph or table representation of control, data flow, or state transition structure, or in documentation of the product itself.

6.1.3 Quality in use metrics

The objective of software quality is to achieve quality in use. For systems with end users, this means that specified types of users should be able to carry out specific types of tasks to a required level of productivity and user satisfaction in specified environments. Evaluating quality in use validates software quality in specific user-task scenarios.

Quality in use is the user's view of the quality of a system containing software, and is measured in terms of the result of using the software, rather than properties of the software itself. Quality in use is the combined effect of the software quality characteristics for the end user.

Quality in use may be influenced by any of the quality characteristics, and is thus broader than usability, which is only concerned with the ease of use and attractiveness. Quality in use can be measured by the extent to which the specified users can achieve their goals with effectiveness, productivity (task efficiency), safety and satisfaction. Effectiveness can be measured by the accuracy and completeness with which users achieve specified goals, productivity (task efficiency) by the resources expended in relation to task effectiveness, safety by the level of impact with possibility against to specific risk arising scenarios, and satisfaction by attributes to the use of the product. Quality in use metrics should be designed to:

1. describe achieved level of specific types of tasks which are carried out by the specified types of users;

2. validate software quality in specific user-task scenarios.

Quality in use metrics should be designed to employ the measurements of the extent to which the specified users can achieve their goals of users' specific tasks.

NOTE: External usability metrics such a particulary operability and attractiveness one may employ measurements of user behavior during operation, as well as, quality in use metrics. However, There are themajor differences between external usability and quality in use metrics which are the followings:

- usability metrics are intended to give the measures represent the extent of which user support function is avilable or influences is appeared to user behaviour;

-quality in use metrics are intended to give the measures represent the extent of which user can achieve goals of users' specific types of tasks.

6.2 Interpretation of measurement

The purpose of this sub clause is to clarify to the reader the interpretation of the measurement concept according to the ISO/IEC 9126-1 and ISO/IEC 14598-1.

6.2.1 Direct measure for software

A direct measure is a measure of an attribute that does not depend upon a measure of any other attribute.

Some measurements of software attributes can be done without being influenced by factors such as the choice of the server where the software is executed, behavior of a user, or other external factor. Those measurements are referenced here as direct measures. Some examples are size of source and executable code, total number of menu options of an application or number of configuration items.

6.2.2 Indirect measure for software

An indirect measure is derived from (direct or indirect) measures of one or more attribute. For example, measure of response time is not only affected by the evaluated software itself, but also by the operating environment including but not restricted to computer hardware and operating system software. So, the response time of software is derived from the response time of the computing environment as well.

6.2.3 Indicators

Some measures can be estimated or predicted from other measures. Those measures are referenced here as indicators, and may be useful to estimate or predict attributes that cannot be measured directly, or can not be measured at all without a model. For example, the response time is not measurable while the software is still a non-executable intermediate product. Therefore, program path length may be measured and used as an indicator to predict the future response time before the software becomes executable.

As another example, in the case that the software is executed in in-house testing, the response time under a testing environment is measurable but may vary from the response time experienced by an actual user in the final operating environment. Therefore, the response time of the software in the final user environment can be predicted from the response time of the testing environment.

Finally, in the case that measuring efficiency of the software is dominantly dependent on time measurement, the response time may be used as an indicator to represent efficiency quantitatively in the software quality evaluation.

6.3 Metrics desirable properties

The accuracy and correctness of a quality evaluation relies strongly on the metrics used on that process. In order to improve metrics confidence, a list of validation requirements, which should be present on every metric applied in an evaluation, is given below.

Whenever a metric does not satisfy these validation requirements, the metric description should explain the associated constraint and, as far as possible, how that situation can be handled.

NOTE: Those properties are suited to the requirements on measurements of ISO/IEC 14598-1, and to the requirements on the evaluation of ISO/IEC 14598-5.

Reliability (of metric): Reliability is associated with random error. A metric is free of random error if random variations do not affect the results of the metric:

a) Repeatability: repeated use of the metric in the same product to the same evaluation specification (including the same environment) by the same evaluators, test users and environment should produce results that can be accepted as being identical,

b) Reproducibility: use of the metric in the same product to the same evaluation specification (including the same environment) by different evaluators, test users and environment should produce results that can be accepted as being identical.

NOTE: It is recommended to use statistical analysis to mitigate random variations within acceptable range.

Indicativeness (of metric): Capability of the metric to identify parts or items of the software which should be improved, given the measured results compared to the expected ones.

Availability (of metric): The metric should make it clear the conditions (e.g.: presence of specific attributes) which constrain its usage.

NOTE: The selected or proposed metric should provide documented evidence of the availability of the metric for use.

Cost Effectiveness (of metric): The metric should have a good cost-benefit relation, that is, the most expensive its application, the most important the results obtained. NOTE : As an example, metrics which demand users and hardware for testing could be more expensive than those requiring project inspection only.

Correctness (of metric): The metric should have good property which encompasses objectivity, impartiality and precision.

a) Objectivity: the metric results and its data input should be factual, i.e. not colored by the feelings or the opinions of the evaluator, test users, etc.

NOTE: Since it is impossible to avoid subjective factors, especially on usability evaluation, it is recommended that procedure for assigning the number or category is enough well reviewed to be agreeable. This does not mean that it is not applicable such a metric with user sensitive testing based on questionnaire or interview.

b) Impartiality: the measurement should not be biased towards any particular result.

c) Precision: Precision is determined by the design of the metric, and particularly by the choice of the material definition used as the basis for the metric. The metric user will describe the precision and the sensitivity of the metric.

Meaningfulness (of metric): the measurement should produce meaningful results about the software behavior or quality characteristics. User of metrics will describe the goal or question the metric result aims at fulfilling.

7 Basic use of external metrics for quality characteristics (Metrics tables)

The following set of metrics is recommended to be used as basic metrics which gives measures or may be applied as checklists to represent software quality characteristics. Although there are some practical experiences in progress, these metrics are draft and need more validation and feedback. They are listed in order of software quality characteristics and subcharacteristics.

Additional specific metrics for particular purposes are not included, but are provided in other related documents, such as functional size measurement or precise time efficiency measurement.

Metrics, which may be applicable, are not limited to these listed here. If there is a trial to apply a new metric, the appropriate model and the practical experiences should be evaluated and specified.

NOTE : It is recommended to refer a specific metric or measurement form specific standards, technical reports or guidelines Functional size measurement is defined in ISO/IEC 14143s. An example of precise time efficiency measurement can be refered from ISO/IEC 14756. For critical applications, it is recommended to use other specific standards that give precise guidance on how to achieve and measure the required critical quality characteristics such as safety, security, and human factors.

The title terms used in these metrics tables are followings.

a) Metrics name : Metrics name characterizes a measureable attribute of software and represents an unique or a group of measurements. Metrics name has the same or similar name in internal, external and quality in use metrics, when they are intended to be mutually corresponded respectively.

b) Purpose of the metrics : This helps to identify what user of metric can know by using the metric. This is described as a questionary style to look up easily in accordance with Goal / Question / Metric flamework.

c) Method of application : This helps to understand what way are useful and recommended to apply metrics.

d) Measurement, formula and data element computations : This helps to understand what kind of measurement, formula and data element are used to compute measure.

NOTE: In many situations one or more measurement formulas for one metric is proposed. That is why a lot of metrics can employ some measurements such are counting up the result with checked each function item or counting up occurrences of problems on the executed software or on user behavior during operational testing. It is recommended to regard metric has not a single number or category, but a data structure.

e) Interpretation of measured value : This helps to understand the range of measured value and the interpreted better range.

f) Metric scale types : The measurement metric scale types should be identified for each measure, when a user of ıetrics has the result of a measurement and use measure to calculate together or compared with other. The average, ratio or difference values may have no meaning for some measures. Such scale types are: Nominal scale, Ordinal scale, Intervals scale, Ratio scale, Absolute scale. M'=F(M), where F is the admissible function, explain what the admissible function is (if M is a metric then M'=F(M) is also a metric).

NOTE :
1. Measurement (metric) scale type categories measurement mapping , from an observed (empirical) relation system to some numerical relation system. Measurement (metric) scale type does not depend on its calculation formula.

2. As an appropriate statistics, Mode and Frequency can be used for Nominal type, Median and Percentile for Ordinal type, Mean and Standard deviation for Interval type, and Geometric mean and Coefficient of variation for Ratio type. All arithmetic analysis of resulting count is meaningful for Absolute type.

3. Be aware line of code is Ratio type, because there are many different way to measure it(such as LOC, number of characters, and number of bytes). A count of the number of failure and the number of people working on a software project are Absolute type, because they can be measured only in one way.

g) Measurements types : For designing procedure for collecting data, interpreting fair meanings, and normalizing measures for comparison, a user of metrics should identify and take account of measure type of measurement mployed by a metric.

Size measure type : A measure of this type represents a particular size of software according what it claims to measure within its definition. Normalizing other measures with a size measure can give comparable values in terms of units of size. The size measures in this report can be used for software quality metrics. Functional size, as well as source lines of code, is an example of one type of size that software may have.

Time measure type : The user of metrics of time measure type should record time periods, how many examined sites and how many users took part of measurements. The user of metrics should be aware that there are many ways in which time can be measured as a unit.

Count measure type : This measure type identifies number of counted number, turn, event or incident with investigation activity. Investigation activity includes reviewing, testing and operating, and using from view of human-engineering and ergonomics.

h) Input to measurement : This helps to understand what kinds of information or documents are generally required to do measurement.

i) ISO/IEC 12207 SLCP Reference : This suggest processes of software life-cycle processes (SLCP) defined in ISO/IEC 12207, in which the metric is beneficially applicable.

j) Beneficeries : This helps to understand whose view and benefit are strogly related with the metric, though any other personel and party related to the software are also receive benefit.

NOTE:

1. All the title terms used in these tables are described farther more in annex A.

2. All the definitions related to this clause are presented in ISO/IEC 9126-1.

3. All metrics described in this clause are related to the following:

The behavior of the system may be observed with following aspects:

a) differences between actual executed results and quality requirements specifications (a view of quality validation testing);

b) occurrences of unexpected time behavior or resource utilization during operation, which may be due to not stated implied needs.

7.1 Functionality metrics

An external functionality metric should be able to measure an attribute such as behavior of system containing the software.The behavior of the system may be observed with following aspects:

a) differences between actual executed results and quality requirements specification (a view of quality in use validation testing);

NOTE : Quality requirements specification for functionality is usually described as functional requirements specification.

b) insufficient performed function during actual user operation that may be implied not stated (a view of quality in use). NOTE : When a lot of implied operation or function are detected, they should be again reviewed, approved and stated. Their extent to be fullfilled should be agreed.

7.1.1 Suitability metrics

An external suitability metric should be able to measure an attribute such as the occurrence of the unsatisfied functions or occurrence of unsatisfied operations from behavior of system during testing and user operation.

Unsatisfied function implies:

a) a function performing task which does not conform to specified ones in requirements specifications or user manuals, or functions specified in the user manual that are simply not performed;b) a function, which does not meet appropriately the tasks for user's reasonable, intended specific use.

7.1.2 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 calculation results or less precision results beyond the range of erroneous of calculation, for example, it is caused why it is too shortage of significance to calculate;

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.

7.1.3 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 is transferred easily between the software product and other systems, other software products, or equipment which are connected.

7.1.4 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:

- failing to prevent leak of secure output information or data;

failing to prevent lost of important data;

- failing to defend against illegal access or illegal operation.

NOTE : 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 as "quality in use".

7.1.5 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 is of the software product to failing to adhere to standards, conventions or regulations in laws and similar prescriptions which are required to be adhered.

Table 7.1.1 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	Beneficieries
Functional implementation completeness	How much 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.	$X = 1 - (A / B)$ 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	Requirement specification Evaluation report	6.5 Validation, 6.3 Quality Assurance, 5.3 Qualification testing	Developer, SQA
	NOTE: 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.			NOTE : 2. This metric is suggested as experimental use.					
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 functions that are complete versus the ones that are not.	$X=1-(A/B)$ 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	Requirement specification Evaluation report	6.5 Validation, 6.3 Quality Assurance, 5.3 Qualification testing	Developer, SQA
	NOTE: 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.			NOTE : 2. This measure represents a binary gate checking of determining whether presence or absence of a feature.					

Functional specification stability (volatility)	How much stable are the functional specifications of the system right after entering operation ?	Count the number of functional specifications that had to be changed after the system is put into operation	$X = 1 - A / B$ A= Number of functions changed after entering operation starting from entering operation B= Number of functions described in requirement specifications NOTE : This metric is suggested as experimental use.	$0 <= X <= 1$ The closer to 1.0 is the better.	Absolute.	A= Count B= Count X= Count/ Size	Requirement specification Evaluation report	6.8 Problem Resolution 5.4 Operation	Maintainer

Table 7.1.2 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 measurement	ISO/IEC 12207 SLCP Reference	Beneficieries
Accuracy to expectation	Are differences allowable between the actual and reasonable expected results ?	Do input .vs. output test cases and compare the output to reasonable expected results. Count the number of test cases beyond allowable difference against reasonable expected results.	X=A / T A= Number of users encountered cases beyond allowable difference against reasonable expected results T= Operation time NOTE : Reasonable expected results may be identified in a requirement specification, a user operation manual, or hearing to users	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
Computational Accuracy	How often does 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 does 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

Table 7.1.3 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 measurement	ISO/IEC 12207 SLCP Reference	Beneficeries
Data exchangeability (Data format based)	How complete are the downstream interface functions for specified data transfer ?	Test each downstream interface function output record format of the system according to the data fields specifications.	Data exchangeable format $X = A / B$ A= Number of data formats which are approved to be exchanged with other software or system during testing on data exchanges, B= Total number of data formats to be exchanged NOTE : It is recommended to test specified data transaction.	$0 <= X <= 1$ The closer to 1.0 is the better.	Absolute.	A=Count B=Count X=Count/Count		6.5 Validation	Developer
Data exchangeability (User's success attempt based)	How often successful are the data transfers between target software and the other software ? Can user usually succeed to exchange data ?	Count the number of times that an interface functions was used and failed.	User successful data exchange ratio $Y = 1 - (A / B)$ A= Number of cases which user fail to exchange data with other software or systems B= Number of cases which user attempt to exchange data	$0 <= Y <= 1$ The closer to 1.0 is the better.	Absolute.	A=Count B= Count Y=Count/Count		5.4 Operation	Maintainer

Table 7.1.4 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	Beneficeries
Access auditability	How complete is the audit trail concerning the user access to the system and data ?	Evaluate the amount of access record that the system records in the access history database.	Access history record holding ratio 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 NOTE :1. Accesses to data may be measured only with testing activities. 2. This metric is suggested as experimental use. 3. 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 as "quality in use"	0<=X<=1 The closer to 1 is the better.	Absolute.	A=Count B=Count X=Count/Count	Test spec. Test report	6.5 Validation	Developer
Access controllability	How well is access to the system controlled?	Try to get access to the system by unauthorized ways.	Illegal operation access detectable ratio X= A / B A= Number of detected illegal operations B= Number of illegal operations anticipated in specification NOTE : If it is necessary to complement unexpected illegal operations to be detected, additional intensive abnormal operation testing should be conducted.	0<=X<=1 The closer to 1 is the better.	Absolute.	A=Count B=Count X=Count/Count	Test spec. Test report Operation report	6.5 Validation 5.4 Operation 6.3 Quality Assurance	Developer

Table 7.1.4 Security metrics (continued)

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	Beneficeries
Data corruption Prevention	How often fatal data corruption occur ?	Count the occurrences of data corruption, and of fatal data coruptions events.	Frequency of fatal data corruption events X= 1 – A / N A= Number of times that a major data corruption event occurred N= Number of test cases tried to occur data corruption event Frequency of data corruption events Y= 1- B / N B= Number of times that a minor data corruption event occurred NOTE : 1. Intensive abnormal operation testing is needed to obtain minor and major data corruption events. 2. It is recommended to grade impact of data corruption event such are following examples. 3. Major data corruption event : - reproduce and recover impossible ; - second affection distribution to wide ; - importance of data itself. B) Minor data corruption event : - reproduce or recover possible and – no second affection distribution ; - importance of data itself. 4. This metric is suggested as experimental use.	0<=X<= 1 The closer to 1.0 is the better.	Absolute.	A=Count B=Count X=Count/ Count Y= Count/ Count	Test spec. Test report Operation report	6.5 Validation 5.3 Qualification testing 5.4 Operation	Maintainer

Table 7.1.5 Compliance metrics (compliance for functionality)

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	Beneficiaries
Functional compliance	How completely is compliance to applicable regulations, standards and conventions being satisfied ?	Previously specify required compliance items based on standards, conventions or regulations in laws which to be adhered by software. 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.	Ratio of satisfied compliance items $X = 1 - (A / B)$ A= Number of failed compliance items during testing B= Number of total compliance items NOTE : 1. It may be useful to collect several measured values along time, to analyse the trend of increasing satisfied compliance items and to determine whether they are fully satisfied or not. 2. It is suggested to count number of fail, because problem detection is an objective of effective testing and also suitable for counting and recording.	$0 <= X <= 1$ The closer to 1 is the better.	Absolute.	A= Count B= Count X= Count/ Count	Specification of compliance and related standards, conventions or regulations Test specification and report	5.3 Qualification testing 6.5 Validation	Supplier User
Intersystem standard consistency	How consistent are the standards in the systems concerned?	Test each interface of the system for consistency with the standard identified in the specifications.	Interface standard consistent ratio $X = A / B$ A= Number of check items of intersystem interface which are approved at testing that they are consistent with standard/rule of intersystem. B= Total number of check items of intersystem interface	$0 <= X <= 1$ The closer to 1.0 is the better.	Absolute.	A=Count B=Count X=Count/ Count		6.5 Validation	Developer

7.2 Reliability metrics

An external reliability metric should be able to measure attributes related to the behaviors 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 case.

7.2.1 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.

7.2.2 Fault tolerance metrics

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

7.2.3 Recoverability metrics

An external recoverability metric should be able to measure such attribute as the software with system being able to re-establish its adequate level of performance with minimal lost of data.

7.2.4 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, which is of the software product to failing to adhere to standards, conventions or regulations relating to reliability which are required to be adhered.

Table 7.2.1 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 measurement	ISO/IEC 12207 SLCP Reference	Beneficeries
Estimated latent failure density	How many problems still potentially exist and may emerge ? How often may users come across failures or faults in the future ?	Count the number of detected failures and predict potential number by using a reliability growth estimation model.	Estimated residuary latent failure density X= {ABS(NPFI - NAFI)} / SIZE ABS()= Absolute Value NPFI= total number of predicted latent failures NAFI= total number of actually detected failures SIZE= product size	0<=X It depends on stage of testing. Finally, the smaller is the better.	Ratio	NPFI= Count NAFI= Count SIZE= 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
	NOTE : 1.When total number of actually detected failures becomes larger than total number of predicted latent failures, it is recommended to predict again and estimate more larger number. 2. It is recommended to use several reliability growth estimation models and chose the most likelihood fit one and repeat prediction with monitoring detected failures.			NOTE : 3. It may be helpful to predict upper and lower number of latent failures. 4. It is necessary to convert this value (X) to the <0,1> interval if making summarization of characteristics					
Estimated latent fault density	How many problems still potentially exist and may arise failures ?	Count the number of detected faults and predict potential number by using several reliability growth estimation models.	Estimated residuary latent fault density X= {ABS(NPFU - NAFU)} / SIZE ABS()= Absolute Value NPFU= total number of predicted latent faults in a software product NAFU= total number of actually detected faults SIZE= product size	0<=X It depends on stage of testing. Finally, the smaller is the better.	Ratio	NPFT= Count NAFT= Count SIZE= Size X= Count/ Count	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
	NOTE : 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. 2. It is recommended to use several reliability growth estimation models and chose the most likelihood fit one and repeat prediction with monitoring detected faults.			NOTE : 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 summarization of characteristics					

Table 7.2.1 Maturity metrics (continued)

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	Beneficeries
Failure density **(Fault density)**	How many problems were detected ?	Count the number of detected failures (or faults) and calculate density.	a) Failure density X= NFAI / SIZE b) Fault density Y= NFAU / SIZE NFAI= number of detected failures NFAU= number of detected faults SIZE= product size	0<=X,Y It depends on stage of testing. Finally, the smaller is the better.	Ratio	NFAI= Count NFAU= Count SIZE= Size X,Y= 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

NOTE : 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 trend of this measure along with time.
2. The number of detected failures (or faults) divided by the number of test cases indicates effectiveness of test cases.

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

Table 7.2.1 Maturity metrics (continued)

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	Beneficeries
Failure Resolution	Evaluate the amount of failure conditions that are resolved.	Failure resolution is ensured by observation that the same type of failure never occurs again with the execution condition being acceptably identical to the condition with which failure once occurred. Maintain a problem resolution report describing status of all the failures.	a) Ratio to actually detected number X= NRFI / NAFI NRFI= number of resolved failures NAFI= total number of actually detected failures b) Ratio to estimated number Y= NRFI / NPFI NPFI= total number of predicted latent failures	0<=X<= 1 The closer to 1.0 is the better. More failures resolved 0<=Y The closer to 1.0 is the better.	a)Absolute. b) Absolute	NRFI= Count NAFI= Count NPFI = Count X= Count/ Count Y= Count/ Count	Test report Operation (test) report	5.3 Integration 5.3 Qualification testing 5.4 Operation	Maintainer

NOTE:
1. It is recommended to monitor trend of this measure along with time.
2. Total number of predicted latent failures may be estimated with one of reliability growth models adjusted by parameter data based on the past history.

3. It is recommended to monitor this estimated failure resolution ratio Y andwhen Y > 1, to investigate whether it is going well to detect and to resolve early more enough defects before delivery than estimation, or it is going worse because of more defects contained in the software than usual.

Otherwise, when Y < 1, then it is recommended to investigate whether it is going well because of more defects contained in the software than usual, or it is going worse because of defects detected less than usual.

4. Multiple problem reports may be received for the same failure, when the software is under user beta testing or multiple sites testing.

NOTE:
5. It is necessary to convert this value (Y) to the <0,1> interval if making summarization of characteristics

Table 7.2.1 Maturity metrics (continued)

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	Beneficeries
Fault Removal	How many faults have been corrected ?	Count the number of detected faults and permanently corrected faults. Fault correction is ensured by observation of resolved failure or by testing or reviewing modification. Maintain a problem resolution report describing status of all the failures.	a) Ratio to actually detected number X= NCFU / NAFU NCFU= number of corrected faults NAFU= total number of actually detected faults b) Ratio to estimated number Y= NCFU / NPFU NCFU= number of corrected faults NPFU= total number of predicted latent faults in a software product	0<=X<= 1 The closer to 1.0 is the better, less faults remain. 0<=Y The closer to 1.0 is the better, less faults shall remain	(a)Absolute. (b) Absolute.	NCFU= Count NAFU= Count NPFU= 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

NOTE:

1. It is recommended to monitor trend of this measure along with time.

2. Total number of predicted latent faults may be estimated with one of reliability growth models adjusted by parameter data based on the past history.

3. It is recommended to monitor this estimated faults resolution ratio Y and
when Y > 1,
to investigate whether it is going well to detect and to resolve early more enough defects before delivery than estimation, or it is going worse because of more defects contained in the software than usual.

Otherwise, when Y < 1,
then it is recommended to investigate whether it is going well because of more defects contained in the software than usual, or it is going worse because of defects detected less than usual.

NOTE:

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

Table 7.2.1 Maturity metrics (continued)

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	Beneficeries
Mean time between failures (MTBF)	How frequent are the failures of the system in operation?	Trace failures occurred during observed testing or operation time.	Mean time between failures a) X = TOPT / NAFI b) Y = TSIB / NAFI TOPT = operation time TSIB = sum of time interval between failure occurrence and the next one NAFI = total number of actually detected failure (Failures occurred during observed operation time)	0<X,Y The longer is the better. The longer time can be expected between failures.	(a) Ratio (b) Ratio	NAFI= Count TOPT= Time TSIB= 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
	NOTE: 1. The following investigation may be helpful: - distribution of time interval between failure occurrence to the next; - 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.			NOTE: 2. Failure rate or hazard rate calculation may be alternatively used. 3. It is necessary to convert this value to the <0,1> interval if making summarization of characteristics					

Table 7.2.1 Maturity metrics (continued)

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	Beneficeries
Test coverage	Have enough user operation scenario test cases been executed?	Count the number of test cases from user operation view which are actually tested.	Specified operation scenario testing coverage X= A / B A= Number of actually performed test cases representing operation scenario during testing B= Number of test cases to be performed NOTE: 1. Test cases may be normalized 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.	0<=X<=1 The closer is to 1, the better test coverage.	Absolute.	A= Count B= Count X= Count/ Count	Test spec. Test report Operation report	5.3 Qualification testing 6.5 Validation 6.3 Quality Assurance	Developer Tester SQA
Test overcome	Is product going to pass successfully a lot of test cases?	Count the number of passed test cases which have been actually executed.	Passed test case ratio X= A / B A= Number of passed test cases during testing or operation B= Number of test cases to be performed. NOTE: 1. It is recommended to perform heavily stress testing, for example, to use actual log data at highly peak period of operation for testing. Therefore, it is recommended to ensure the following test cases are executed enough and passed successfully: - User operation scenario; - Peak stress; - Overloaded data input. 2. Passed test cases may be normalized by	0<=X<=1 The closer is to 1 the better overcome of tests.	Absolute.	A= Count B= Size X= Count/ Size	Test spec. Test report Operation report	5.3 Qualification testing 6.3 Quality Assurance	Developer Tester SQA

software size, that is:
passed test case density $Y = A / C$, where
C= Size of product to be tested.
The larger Y is the better.
Size may be functional size that user can measure.

Table 7.2.2 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 measurement	ISO/IEC 12207 SLCP Reference	Beneficieries
Breakdown avoidance	How often can user avoid breakdown of system, even if critical failures occurred?	Count the number of breakdowns occurrence with respect to number of failures. If it is under operation, analyse log of user operation history.	Breakdown avoidance ratio X= 1-(A / B) A= Number of breakdowns B= Number of failures NOTE: 1.The breakdown means executing of any user task is suspended until system is restarted up, or its control is lost until system is enforced to be shut down. 2. When none or a few failures observed, time between breakdown may be more suitable.	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 Qualifica-tion testing 5.4 Operation	User Maintainer

Fault tolerance metrics (continued)

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	Beneficeries
Failure avoidance	How often can user avoid critical or serious failure?	Count the number of critical and serious failure occurrence with respect to number of causes.	X=A / B A= Number of avoided critical and serious failure emergence 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 is the better, 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
	NOTE: 1. It is recommended to categorize 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 out of service and no alternative way of operating (workaround); -Average: most functions are still available, but limited performance occur with limited or alternate operation (workaround); -Small: a few functions experience limited performance with limited operation; -None: impact does not reach end user			NOTE: 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 case can include the human incorrect operation					
Incorrect operation avoidance	How often can user avoid failure, even if user operates incorrectly?	Do operational test or analyse log of user operation history. Count the number of critical and serious failure occurrence with respect to number of user's incorrect operation.	X=A / B A= number of avoided user's erroneous operations or inputs during operation B= number of user's erroneous operation or input during operation NOTE: This metric may be used experimentally.	0<=X<= 1 The closer to 1.0 is better, more incorrect user operation avoided	Absolute.	A=Count B=Count X=Count/ Count	Test report Operation report	5.3 Integration 5.3 Qualification testing 5.4 Operation	User Maintainer

Table 7.2.3 Recoverability 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	Beneficeries
Availability	Can user use software system enough every time?	Investigate recovery testing.	(a) Availability X= { To / (To + Tr) } (b) User operation available ratio Y= C / D (This is from software system operation view.) To = operation time Tr = time to repair C= total available turns of user's successful software use when user attempt to use D= total number of turns of user's attempt to use the software during observation time. This is from user callable function operation view. NOTE: It is recommended that this metric includes only the automatic recovery provided by the software and excludes the maintenance work of human.	0<=X<=1 The larger is the better, the user can use the software for more time. 0<=Y<=1 The larger is the better.	(a),(b) Absolute.	To = Time Tr = Time X= Time/ Time C= Count D= Count Y= Count/ Count	Test report Operation report	5.3 Integration 5.3 Qualifica-tion testing 5.4 Operation	User Maintainer
Mean down time	How long is usually system down?	Investigate recovery testing.	Mean time: X= T / B T= Total down time B= Number of observed breakdowns The worst case or distribution of down time should be measured.	0<X The smaller is the better, system will be down for shorter time.	Ratio	T=Time B=Count X=Time/Count	Test report Operation report	5.3 Integration 5.3 Qualifica-tion testing 5.4 Operation 6.5 Validation	User Maintainer

NOTE:
1. It is recommended that this recovery metric includes only the automatic recovery provided by the software and excludes the maintenance work of human.

NOTE:
2. It is necessary to convert this value (X) to the <0,1> interval if making summarization of characteristics

Table 7.2.3 Recoverability metrics (continued)

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	Beneficeries
Recovery	How long it will take to recover if system went down?	Let system to be down and compare estimated time with actual time to re-establish its adequate level of performance with enough less lost of data	Recovery time Mean time =X= Sum(T) / B T= Time to recovery downed software system at each opportunity B= Number of turns which observed software system downs entered into recovery	0<X The smaller is the better.	Ratio	T=Time B=Count X=Time/ Count	Test report Operation report	5.3 Integration 5.3 Qualification testing 5.4 Operation 6.5 Validation	User Maintainer
	NOTE: 1. It is recommended to measure maximal time of the worst case or distribution of recovery time for many cases. 2. It is recommended that this recovery metric includes only the automatic recovery provided by the software and excludes the maintenance work of human.			3. It is recommended to distinguish the grades of recovery difficulty, for example, recovery of destroyed data base is more difficult than recovery of destroyed transaction. 4. It is necessary to convert this value (X) to the <0,1> interval if making summarization of characteristics					
Restartability	Can user restart easily?	Let system to be down, and compare estimated time with actual time to restart system.	X = A / B A= Number of restarts which met to required time during testing or user operation support B= Total number of restarts during testing or user operation support	0<=X<=1 The larger and closer to 1.0 is better, the user can restart easily.	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
	NOTE: 1. It is recommended to estimate different time to restart to correspond to the severity level of down, such as data base destruction, lost multi transaction, lost single transaction, or temporal data destruction. 2. It is recommended that this recovery metric includes only the automatic recovery provided by the software and excludes the maintenance work of human.								

Table 7.2.3 Recoverability metrics (continued)

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	Beneficeries
Restorability	How is the product capable to restore in defined cases ?	Let the system to perform defined cases	X= A / B A= Number of cases successfully restored B= Number of cases performed NOTE: It is recommended that this metric includes only the automatic recovery provided by the software and excludes the maintenance work of human.	0<=X<=1 The larger is the better, the product is more capable to restore in defined cases	Absolute	A= Count B= Count X= Count/ Count	Test report Operation report	5.3 Integration 5.3 Qualifica-tion testing 5.4 Operation 6.5 Validation	User Maintainer
Restore effectiveness	How effective is the restoration process in the product ?	Let the system to perform defined cases	X= A / B A= Number of cases successfully restored meeting the target restore time B= Number of cases performed NOTE: It is recommended that this metric includes only the automatic recovery provided by the software and excludes the maintenance work of human.	0<=X<=1 The larger is the better, the restoration process in product is more effective.	Absolute	A= Count B= Count X= Count/ Count	Test report Operation report	5.3 Integration 5.3 Qualifica-tion testing 5.4 Operation 6.5 Validation	User Maintainer

Table 7.2.4 Compliance metrics (compliance for reliability)

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	Beneficiaries
Reliability compliance	How completely does the software adhere the standards, conventions or regulations relating to reliability?	Previously specify required compliance items based on standards, conventions or regulations relating to reliability which to be adhered by software. Design test cases in accordance with compliance items. Conduct functional testing for these test cases.	Ratio of satisfied compliance items relating to reliability X = 1 – (A / B) A= Number of failed compliance items during testing B= Number of total compliance items NOTE: It may be useful to collect several measured values along time, to analyse the trend of increasing satisfied compliance items and to determine whether they are fully satisfied or not.	0<= X <=1 The closer to 1 is the better.	Absolute.	A= Count B= Count X= Count/ Count	Specification of compliance and related standards, conventions or regulations Test specification and report	5.3 Qualification testing 6.5 Validation	Supplier User

7.3 Usability Metrics

External usability metrics should be able to measure software attributes related to its operation, with regard to easiness of use and adaptation of new operators. Although the evaluation focus is the software, it's expected to be difficult to distinguish between actual software attributes, and the host system influence over usability characteristics. This does not invalidate the measurements, since the evaluated software is ran under explicitly specified conditions, which encompass the required system."

An external usability metric should be able to measure such attributes as user behavior, operator or system including software, to represent the extent of ease of use.

NOTE: User tests

Most external usability metrics are based on testing users. A sample of users who are representative of an identified user group should carry out the test without any hints or external assistance. These measures may vary widely among different individuals. For reliable results a sample of at least eight users is necessary, although useful information can be obtained from smaller groups.

It should be possible for the measures taken to be used to establish acceptance criteria or to make comparisons between products. This means that the measures should be counting items of known value. Results should report the mean value and the standard error of the mean.

7.3.1 Understandability metrics

Understandability metrics should be capable of evaluating the behavior of users without previous knowledge on software operation and measure their difficulty on understanding software functions, operation and concepts. On doing this, it may be considered entities such as documentation (in all available formats, as on-line or printed), software interface and vocabulary. The condition of observing users without any background on the evaluated software is not always neccesarry, but is expected to be helpful on measuring the capability of the software to communicate with new users; at that point, usability problems are more related to software problems then to user psychological aspects, as stress on trying to operate the system.

An external understandability metric should be able to measure such attribute as users' effort by measuring the behavior of user who, before regularly use of software, try to understand:

a) what kind of tasks are performed or outputs produced by the software product;

b) what kind of users' manual tasks, operations, or inputs are needed to use the software.

These assist users to select the software product which is suitable to their intended use.

7.3.2 Learnability metrics

Learnability metrics should be capable of evaluating or drawing the user curve of performance on software operation, from a start point of no knowledge about the evaluated software.

The main measurable factors are time and success ratio on completing tasks by using software operations. When using prepared or chosen test tasks, the evaluator must be aware of the effects of inducing user behavior, like ordering those tasks in an increasing difficulty list.

Learnability measuring is strongly related to understandability; in the evaluation results analysis, and during the evaluation process, understandability measurements probably can be indicators of the learnability potential of the evaluated software."

An external learnability metric should be able to measure such attribute as the behavior of user who is learning how to use the software. Measurement may include measures derived from interviewing to user.

7.3.3 Operability metrics

An external operability metric should be able to measure such attribute as user's human behavior during operational testing, usability testing or user operation.

The followings should be remarked:

a) Operation testing should be performed and operation of users should be observed and analyzed;

b) The software or functions which are specified to be used infrequently or intermittently could be excluded (depending of the nature of the needs). Examples are emergency call function or restart function;

c) Frequency distribution for each of functions may be helpful to understand user's operational profile.

NOTE:1. When user's operation profile is comprehended, it is useful to apply usability testing to frequently used function with priority derived from profile, and to get close operability measurement results to user's actual evaluation.

2.Users may be observed who are participants of operation testing.

7.3.4 Attractiveness metrics

An external attractiveness metric should be able to measure such attribute as user's human behavior expressing the extent of which user likes the software during operational testing, usability testing or user operation.

7.3.5 Compliance metrics

An external usability compliance metric should be able to measure an attribute such as the number of functions with, or occurrences of compliance problems, which is of the software product to failing to adhere to standards, conventions, style guides or regulations relating to usability which are required to be adhered.

Table 7.3.1 Understandability 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	Beneficeries
Completeness of description	What proportion of functions (or types of function) are understood after reading the product description?	User test	X = A / B A = Number of functions (or types of functions) understood B = Total of number of functions (or types of functions)	0<=X<= 1 The closer to 1.0 is the better.	Absolute.	A=Count B=Count X=Count/ Count	Operation (test) report	5.3 Qualification testing 5.4 Operation	User Maintainer
Demonstration Availability	Whenever can user see demonstration?	Observe user behavior who is trying to see demonstration. Observation may employ human cognitive action monitoring approach with video camera.	User's demonstration success ratio X = A / B A= Number of turns which user successfully see demonstration when user attempts to see demonstration. B= Number of turns which user attempts to see demonstration during observation period.	0<=X<= 1 The closer to 1.0 is the better.	Absolute.	A=Count B=Count X=Count/ Count	Operation (test) report User monitoring record (video tape and action record)	5.3 Qualification testing 5.4 Operation	User Maintainer
Evident functions	What proportion of functions (or types of function) can be identified by the user based upon start up conditions?	User test	X = A / B A = Number of functions (or types of functions) identified by the user B = Total of number of actual functions (or types of functions)	0<=X<= 1 The closer to 1.0 is the better.	Absolute.	A=Count B=Count X=Count/ Count	Operation (test) report	5.3 Qualification testing 5.4 Operation	User Maintainer
Function understandability	What proportion of interface functions are understandable?	User test	X= A / B A= Number of interface functions whose purpose is correctly described by the user B= Number of functions available from the interface	0 <= X <= 1 The closer to 1.0, the better.	Absolute	A=count B=count X=Count/ Count	Operation (test) report	5.3 Qualification testing 5.4 Operation	User Maintainer
Understandable Input and Output	Is easy for user to understand what is required as input data and what is provided as output by software system?	Observe user behavior who is trying to understand input and output of the software.	Understandable input and output data items ratio X= A / B A= Number of input and output data items which user success to understand during enough short time within a few trial use B= Number of input and output data items with which user attempts to understand during observation period	0<=X<= 1 The closer to 1.0 is the better.	Absolute.	A=Count B=Count X=Count/ Count	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualification testing 5.4 Operation	User Maintainer

Table 7.3.2 Learnability 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	Beneficeries
Ease of function learning	How long does the user take to learn to use a function?	User test	Time required to learn function T= Mean time taken to learn to use a function correctly NOTE: This metric is generally used as one of experienced and justified.	0<T The shorter is the better.	Ratio	T=Time	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualification testing 5.4 Operation	User Maintainer
Ease of performing task learning	How long does the user take to learn how to perform the specified task smoothly ?	Observe user behavior from they started to learn till became to operate smoothly.	Time required to learn operation to perform task T= Sum of user operation time until user achieved to perform the specified task within a short time.	0<T The shorter is the better.	Ratio	T=Time	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualification testing 5.4 Operation	User Maintainer
	NOTE:1. It is recommended to determine an expected user's operating time as a short time. Such user's operating time may be the threshold, for example, which is 70% of time at the first use as the fair proportion.			NOTE: 2. Effort may alternatively represent time by person-hour unit.					
Ease of use of help system	Can the user find online help?	User test	X = A/B A = Number of tasks for which correct online help is located B = Total of number of tasks tested	0<=X The closer to 1.0 is the better.	Absolute.	A= Count B= Count X= Count/ Count	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualification testing 5.4 Operation	User Human interface designer

Table 7.3.2 Learnability metrics (continued)

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	Beneficeries
Effectiveness of help system	What proportion of tasks can be completed correctly after using the help systems?	User test	X= A / B A= Number of tasks successfully completed after accessing online help B = Total of number of tasks tested	0<=X The closer to 1.0 is the better.	Absolute	A= Count B= Count X= Count/ Count	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualification testing 5.4 Operation	User Human interface designer
Effectiveness of user documentation and help systems	What proportion of functions can be used correctly after reading the documentation or using help systems?	User test	X = A / B A = Number of functions that can be used B = Total of number of functions provided NOTE: This metric is generally used as one of experienced and justified metrics rather than the others.	0<=X,Y<=1 The closer to 1.0 is the better.	=Absolute	A= Count B= Count X= Count/ Count	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualification testing 5.4 Operation	User Human interface designer
Tutorial Readiness	Can user easily use tutorial?	Observe user behavior when they try to learn by themselves.	User's tutorial using success ratio X= A / B A= Number of turns which user successfully use tutorial functions when user attempts to use demonstration. B= Number of turns which user attempts to use tutorial functions during observation period.	0<=X<= 1 The closer to 1.0 is the better.	=Absolute	A=Count B=Count X=Count/ Count	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualification testing 5.4 Operation	User Maintainer

NOTE: 1 (Complementary measurement formula)
Function covered by tutorial ratio Y=(C / D)
C= Number of functions of which tutorial functions are used well when user attempts to use.
D= Number of functions of which user attempts to use tutorial functions during observation period.

NOTE: 2 It is recommended to determine previously functions which need tutorials for beginners and to examine later the extent of implementation.

Table 7.3.3 Operability metrics

Conforms with operational user expectations

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	Beneficeries
Operational Consistency	How easily user send his/her intention to or receive his/her expecting results from software through what user see?	Observe user behavior who is operating software	a) Smooth Communicativeness Conformity to WYSIWYG (What You See Is What You Get) X = 1 - (A / B) A= Number of reports or functions which resulted within unacceptable conformity against user's expectation derived from what are visible at the screen B= Number of reports or functions which are visible at the screen	0<=X<=1 The closer to 1.0 is the better.	a) Absolute	A= Count B= Count X= Count/ Count	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualifica-tion testing 5.4 Operation	User Human interface designer
			b) Easy to derive operation Frequency of expected results Y = N / UOT N= Number of operations which resulted within unacceptable conformity against user's expectation derived from experience of operation UOT= user operating time (observation period) NOTE: User's experience of operation is usually helpful to recognize several operation pattern which derives user's expectation.	0<=Y<= 1 The smaller and closer to 0.0 is the better.	b) Ratio	UOT= time Y= Count/ Time			

Operability metrics (continued)

Controllable

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	Beneficieries
Error correction	Can user easily correct error on tasks?	User test	Time for Error Correction on Task T=TCC - TSC TCC = Time completing correct specified type errors of performing task TSC = Time starting correct specified type errors of performing task	0<T The shorter is the better.	Ratio	T=Time			
User operation cancelability	Can user easily recover his/her error or retry tasks?	Observe user behavior who is operating software	a) Frequency of cancel success X= A / UOT A= number of turns which user success to cancel their error operation UOT= user operating time during observation period. NOTE: 1. When function is tested each by each, the ratio can be also calculated, that is the ratio of number of functions which user success to cancel their operation to all functions.	0<=X<= 1 The higher and closer to 1.0 is the better.	Ratio	A=Count UOT = Time X = Count / Time	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualification testing 5.4 Operation	User Human interface designer
	Can user easily recover his/her input?	Observe user behavior who is operating software	b) User's successful input change ratio X = A / B A= Number of screens or forms where the input data were successfully modified or changed before being elaborated B = Number of screens or forms where user tried to modify or to change the input data during observed user operating time	0<=X,Y<=1 The closer to 1.0 is the better.	Absolute.	A= Count. B= Count X= Count/ Count	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualification testing 5.4 Operation	User Human interface designer

Table 7.3.3 Operability metrics (continued)

Suitable for the task operation

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	Beneficieries
Default value availability	Can user easily select parameter values for his/her convenient operation?	Observe user behavior who is operating software. Count how many times user attempts to establish or to select parameter values and fails to do them, because user can not use default values provided by the software.	Frequency of default value available X = 1 - (A / B) A= The number of turns which user fail to establish or to select parameter values in a short period (because user can not use default values provided by the software) B= Total number of turns which user attempt to establish or to select parameter values NOTE: 1) It is recommended to take accounts of operator's behavior and decide how long period is allowable to select parameter values as "short period". 2) When parameter setting function is tested by each function, the ratio of allowable function can be also calculated. 3) It is recommended to conduct functional test which covers parameter setting functions	0<=X<= 1 The closer to 1.0 is the better.	Absolute	A=Count B=Count X=Count/Count	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualification testing 5.4 Operation	User Human interface designer
User operating time adequacy	Is user become to deal task with very short operating time after sufficient training?	Observe user behavior who is operating software before/after sufficiently trained.	a) Ultimate Operating Time Tul = Operating time needed to perform a specified task for ultimately short time skillful or trained user) b) Beginner's Operating Time Tbe = Operating time needed for beginner to perform a specified task c) Reminder's Operating Time Tre = Operating time needed for reminder to perform a specified task	0<Tul 0<Tbe 0<Tre 0<T The shorter is the better.	Ratio	Tul, Tbe, Tre, Tf,Ts and T = Time	Operation (test) report User monitor record	6.5 Validation 5.3 Qualification testing 5.4 Operation	User Human interface designer

d) Operating time needed to complete task
Time to complete a task of user's attempt
$T = Tf - Ts$
Tf = Time completing a specified task
Ts = Time starting operation for task

NOTE:
1. Beginner means that a specified task has been never done by that user or he/she is beginner of the software.

NOTE:
2. Reminder means that task is used only few times with very long intervals and user may forget details of operation.

Operability metrics (continued)

Self descriptive (Guidable)

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	Beneficeries
Guidability	Is everyone adequately guided to success his/her intended task?	Observe user behavior who is operating software Count how many tasks does user attempt and fail/success.	a) Interactive guidability User's attempt task success ratio X= (A / B) A = Number of tasks of which objectives are completed successfully by user with on-line interactive guide B = Number of tasks attempted by user NOTE: It is recommended to investigate the followings with this metric: - Can user complete his/her intended task, even if it is the first attempt to do that? - Can beginner user complete his/her intended task?	0<=X<= 1 The closer to 1.0 is the better.	a) Absolute	A=Count B=Count X=Count/ Count	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualifica-tion testing 5.4 Operation	User Human interface designer
	Is everyone adequately guided to success his/her intended task?	Count how many users fail/success, who are participants of operation testing.	b) Common user guidability Ratio of task success users Y = (C / N) C = Number of users successfully completing task N = Number of users who attempted to do the task NOTE: Common user may be ordinary people or trained operator. Users may be observed who are participants of operation testing.	0<=Y<= 1 The closer to 1.0 is the better.	b) Absolute	C=Count N=Count Y=Count/ Count			

	Observe user's fail or hesitation during operation.	c) User understandable status or progress Frequency of user's fail to understand status or progress Z = D / UOT D= number of turns which user pause for a long period or repeat fail successively at the same operation, because of user not comprehending status or progress of executing task UOT= user operating time (observation period) NOTE: 1. It is recommended to take accounts of operator's behavior and decide how long period is not allowable to pause as "long period". 2. When function is tested each by each, the ratio of allowable function can be also calculated.	0<=Z<= 1 The smaller and closer to 0.0 is the better.	c) Ratio	D = Count UOT= Time Z = Count / Time

Operability metrics (continued)

Operability metrics (continued)

Self descriptive (Guidable)

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	Beneficeries
Message readiness	Can user easily understand messages from software system? Is there any message which brought delay for user to understand and to start next action? Can user easily memorize important message?	Observe user behavior who is operating software	Comprehensive messages Frequency of user's comprehending message X = A / UOT A = number of turns which user pause for a long period or repeat fail successively at the same operation, because of user not comprehending messages UOT = user operating time (observation period)	0<=X<= 1 The smaller and closer to 0.0 is the better.	Ratio	A=Count UOT = Time X = Count / Time	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualification testing 5.4 Operation	User Human interface designer

NOTE:

1. The extent of easiness of message comprehending is represented by how long that message brought delay for user to understand and to start next action. Therefore, it is recommended to take accounts of operator's behavior and decide how long period is not allowable to pause as "long period".

3. It is recommended to investigate the followings as one of the problems of user's comprehending messages.

1)Attentiveness : Attentiveness implies that user successfully recognize attentive important messages presenting such the next user action guidance, name of data items to be looked, and warning for careful operation.

- Can user never fail to watch when user is encountering attentive important messages?
- Can user avoid to mistake operation, because of user's recognizing attentive important messages?

2) Memorability: Memorability implies that user memorize important messages presenting such the next user action guidance, name of data items to be looked, and warning for careful operation.
- Can user easily memorize important message?
- Is it helpful for user to keep user's remembrance?
- Is it required for user to remember only few and not so much?

3. When message is tested each by each, the ratio of comprehensive messages to the total can be also calculated.

4. When several users are observed who are participants of operation testing, the ratio can be calculated, which is ratio of users who comprehended messages to all users.

Operability metrics (continued)

Self descriptive (Guidable)

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	Beneficeries
Self-explanatory error messages	In what proportion of error conditions does the user propose the correct recovery action?	User test	X=A/B A=Number of error conditions for which the user proposes the correct recovery action B=Number of error conditions tested NOTE: This metric is generally used as one of experienced and justified.	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualification testing 5.4 Operation	User Human interface designer

Operability metrics (continued)

Operational error tolerant (Human error free)

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	Beneficeries
Operational error recoverability	Can user easily recover his/her worse situation?	Observe user behavior who is operating software	recoverable situation frequency $X = 1 - (A / B)$ A= Number of unsuccessfully recovered situation (after a user error or change) and user was not informed about this risk by the system B= Number of user errors or changes	$0<X$ The less is the better.	Absolute	A= Count, UOT= Time X= Count/ Time	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualifica-tion testing 5.4 Operation	User Human interface designer
Time Between Human Error Operations	Can user operate the software long enough without human error?	Observe user behavior who is operating software	Time Between Human Error Operations X = Mean time $= T / N$ (at time t during [t-T, t]) T = operation time period during observation (or The sum of operating time between user's human error operations) N= number of occurrences of user's human error operation	$0<X$ The higher is the better.	Ratio	T = Time N = Count X = Time / Count	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualifica-tion testing 5.4 Operation	User Human interface designer

NOTE:
1. Human error operation may be detected by counting below user's behavior :
a) Simple human error: The number of turns which user just simply mistakes to input operation;
b) Intentional error: The number of turns which user repeats fail several times at the same operation with miss-understanding during observation period;
c) Operation hesitation pause: The number of turns which user pause for a long period with hesitation during observation period.
NOTE:
2. It is seemed that operation pause implies user is puzzled and hesitate operation.
It depends on function, operation procedure, application domain and user, whether it is long period or not for user to pause operation. Therefore, evaluator is requested to take account into them and determine threshold time. For an interactive operation, it is generally supposed that "long period" threshold range is from 1min. to 3 min.

Operability metrics (continued)

Operational error tolerant (Human error free)

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	Beneficeries
Undoability	How frequently does the user successfully correct input errors?	User test	a) Input undoability X= A / B A= Number of input errors which the user successfully corrects B= Number of attempts to correct input errors NOTE: This metric is generally used as one of experienced and justified.	0<=X<=1 The closer to 1.0 is the better.	Absolute	A= Count, B= Count X= Count/ Count	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualifica-tion testing 5.4 Operation	User Human interface designer
	How frequently does the user correctly undo errors?	User test	b) Error undoability X= A / B A= Number of error conditions which the user successfully corrects B= Total number of error conditions tested	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualifica-tion testing 5.4 Operation	User Human interface designer

Operability metrics (continued)

Suitable for individualisation

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	Beneficieries
Customisability	Can user easily customize operation procedures for his/her convenience? Can user, who instructs end users, easily set customized operation procedure templates for preventing his/her error? What proportion of functions can be customised?	User test	X= A / B A= Number of functions successfully customised B= Number of attempts to customise	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count			

NOTE:
1. Ratio of user's fail to customize may be measured.
Y = 1 - (C / D)
C = Number of turns which a user fail to customize operation
D = Total number of turns which a user attempted to customize operation for his/her convenience.
0<=Y<= 1, The closer to 1.0 is the better.

2. It is recommended to regard the followings as variations of customise operation:
- chose alternative operation, such as to use menu selection or command input;
- combine user's operation procedure, such as to record and edit operation procedure;
- set constrained template operation, such as to program procedures or to make template for input guidance.

3. This metric is generally used as one of experienced and justified.

Operability metrics (continued)

Suitable for individualisation

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	Beneficieries
Operation procedure reduction	Can user easily reduce operation procedures for his/her convenience?	Count user's strokes for specified operation and compare them between before and after customizing operation.	Operation Procedure Reduction Ratio X = 1 - (A / B) A = Number of reduced operation procedures after customizing operation B = Number of operation procedures before customizing operation. NOTE: 1. It is recommended to take samples for each different user task and to distinguish operator who is skillful user or beginner. 2. Number of operation procedures may be represented by counting operation strokes such click. drug, key touch, screen touch etc.	0<=X< 1 The closer to 1.0 is the better.	Absolute.	A=Count B=Count X=Count/Count	Operation (test) report User monitoring record	6.5 Validation 5.3 Qualification testing 5.4 Operation	User Human interface designer