INTERNATIONAL STANDARD

ISO/IEC 14598-6

First edition
2001-06-01

Software engineering — Product evaluation —

Part 6: Documentation of evaluation modules

Ingénierie du logiciel — Évaluation du produit —

Partie 6: Documentation des modules d'évaluation

Reference number
ISO/IEC 14598-6:2001(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

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

Printed in Switzerland

Contents

Page

Foreword

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

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.

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 14598 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 14598-6 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software engineering*.

ISO/IEC 14598-6 is intended for use in conjunction with ISO/IEC 9126-1 (in preparation) which will replace ISO/IEC 9126:1991.

ISO/IEC 14598 consists of the following parts, under the general title *Software engineering — Product evaluation*:

— *Part 1: General overview*

— *Part 2: Planning and management*

— *Part 3: Process for developers*

— *Part 4: Process for acquirers*

— *Part 5: Process for evaluators*

— *Part 6: Documentation of evaluation modules*

Annexes A, B, C and D of this part of ISO/IEC 14598 are for information only.

Introduction

Software product evaluation depends on a set of evaluation techniques and metrics that provide information about the quality characteristics of the software. Many metrics and associated methods for using the measurement results can be used for specific software product evaluation. ISO/IEC 9126-2 and ISO/IEC 9126-3 provide example metrics that correspond to one sub-characteristic. It is difficult to use these metrics consistently in an organisation. It may be necessary to develop new metrics for specific use. Therefore, it is necessary that a supporting function (see 14598-2) in the organisation specifies each metric for correct and consistent use within the organisation. The format for documenting a metric and associated methods, as well as guides for their use, should be standardised. The concept of an evaluation module provides a solution to this need.

An evaluation module specifies the evaluation methods applicable for evaluating a quality characteristic and identifies the evidence it needs. It also defines the elementary evaluation procedure and the format for reporting the measurements resulting from the application of the techniques.

A consistent way of documenting evaluation modules has a number of advantages:

— It provides a common reference in the description of the theoretical basis of evaluation modules.

— It identifies a minimal set of requirements for documenting and developing evaluation modules.

— It provides a necessary tool in collecting and cataloguing the large number of evaluation modules anticipated.

Evaluation modules provide a flexible and structured approach to making metrics applicable for evaluating intermediate and completed products. The use of evaluation modules produced according to this part of ISO/IEC 14598 helps to ensure that software product evaluations can be repeatable, reproducible and objective.

The format for documenting an evaluation module takes into account the following:

— It will be applied within the context of the evaluation of software products.

— The format supports the need for developing new metrics with respect to state of the art.

— The format provides a precise definition of metrics and their application.

— It provides the information needed for those who will use them for an evaluation.

Annex A provides guidance for the development process for new evaluation modules.

Annexes B, C and D are examples of evaluation modules.

Software engineering — Product evaluation —

Part 6: Documentation of evaluation modules

1 Scope

This part of ISO/IEC 14598 defines the structure and content of the documentation to be used to describe an Evaluation Module. Evaluation modules are intended to be used within the context of the ISO/IEC 9126 and the ISO/IEC 14598 multipart standards.

This part of ISO/IEC 14598 is intended to be used by experts in evaluation technology such as testing laboratories, research institutes and others when producing new evaluation modules.

2 Conformance

The documentation of an evaluation module conforms to this part of ISO/IEC 14598 if it meets the requirements of clause 6 (format for documentation of an evaluation module).

3 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 14598. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 14598 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

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

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

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

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

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

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

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

4 Terms and definitions

For the purposes of this part of ISO/IEC 14598, the following terms and definitions apply.

4.1
evaluation module
a package of evaluation technology for measuring software quality characteristics, sub-characteristics or attributes

NOTE The package includes.

— evaluation methods and techniques

— inputs to the evaluation

— data to be measured and collected

— supporting procedures and tools

4.2
evaluation technology (technology used for evaluation)
the techniques, tools, metrics, measures and other technical information used for evaluation [ISO/IEC 14598-2]

5 The evaluation module concept

Evaluation of software products can be a comprehensive task. Different aspects of quality characteristics and sub-characteristics may require different evaluation techniques to be applied and different data to be collected. In order to manage this complexity, an evaluation should be structured into manageable units. Each of these units can cover one or more quality aspects. However, each unit should be focused on the evaluation of one specific quality aspect applying a specific evaluation technique. The information necessary for conducting one of these evaluations should be collected and packaged for future use. Such a package is called an evaluation module.

The benefit of using a standardized format of an evaluation module is that

— it supports the development of evaluation modules since it provides a table of contents which makes visible what information is necessary for an evaluation and how this information is handled (principles, metrics, tools, ...)

— it supports the use of evaluation modules since information will be available in a homogenous way

— it supports the reuse of evaluation modules since it facilitates the establishment and maintenance of libraries of evaluation modules

— it supports standardization of evaluation modules since the format is compliant with standards requirements

An evaluation module collects all information necessary to perform an evaluation of a specific aspect of a quality characteristic applying a specific evaluation technique. It clarifies which specific aspect of a software quality characteristic is being measured. The procedure for making the measurement is defined as well as the preconditions and accuracy of the measurement.

Evaluation modules provide the link between evaluation techniques, metrics and measures. When parts 3, 4 or 5 of ISO/IEC 14598 recommend the application of an evaluation technique, an appropriate evaluation module can be selected from an evaluation module library (see ISO/IEC 14598-2).

The documentation of an evaluation module has six parts EM0 - EM5 serving different purposes, and an optional annex EMA.

EM0 provides formal information about the evaluation module and gives an introduction to the evaluation technique described in the evaluation module.

EM1 defines the scope of applicability of the evaluation module.

EM2 provides relevant references.

EM3 includes the definitions needed for the evaluation module.

EM4 specifies the input products required for the evaluation and defines the data to be collected and measures to be calculated.

EM5 contains information about how to interpret measurement results.

The optional annex EMA includes the detailed procedure for applying the evaluation module. Although EMA is optional it is suggested that it be included.

NOTE The format for the documentation of an evaluation module complies with the formal requirements of a standard as described in ISO Directives Part 3. This facilitates standardization of evaluation modules.

6 Format for documentation of an evaluation module

The documentation of an evaluation module shall be formatted according to 6.1, 6.2, 6.3, 6.4, 6.5, 6.6 and 6.7.

6.1 EM0 : Foreword and introduction

6.1.1 Foreword

This clause shall provide information about

— preparation, approval, contributions and changes,

— relationship to other standards or other documents.

6.1.2 Introduction

This clause should introduce the principles, background and technical rationale underlying the evaluation module.

NOTE A formal description of evaluation method is provided in 6.2.3.

6.2 EM1 : Scope

6.2.1 Characteristics

This clause shall identify the characteristics, sub-characteristics or attributes that the evaluation module can evaluate.

NOTE The evaluation module may contribute to one or more characteristics or sub-characteristics.

The quality model describing the characteristics, sub-characteristics or attributes shall be identified. The model in ISO/IEC 9126-1 should be used, unless there is a particular reason to use another model.

6.2.2 Level of evaluation

This clause shall describe the level of the evaluation defined by the evaluation module. Evaluation levels are related to the importance of the characteristic, sub-characteristics or attribute evaluated. The level should be

described taking into account the assumed usage of the software and the environment of the software product (for example, safety conditions, security constraints, economic risks, and application constraints).

The level defines the depth or thoroughness of the evaluation in terms of the evaluation techniques to be applied and evaluation results to be achieved. Different evaluation levels give different levels of confidence in the quality of the software product.

NOTE Levels can be formulated as A, B, C, or D as described in ISO/IEC 14598-5. Software integrity levels are described in ISO/IEC 15026.

6.2.3 Techniques

This clause shall describe the evaluation techniques applied by the evaluation module. The relevant theory, models or heuristics underlying the evaluation shall be included or adequately referenced.

NOTE Examples of evaluation techniques are reliability growth models, benchmark testing, static analysis of code.

6.2.4 Applicability

This clause shall identify the scope of application of the evaluation module.

NOTE 1 For example, the evaluation module may be applicable to a particular programming language, or to the class of all imperative languages.

It shall be described where in the software life cycle the evaluation module can be used. If it is intended to be used in a specific software life cycle process, this should be identified.

NOTE 2 Software life cycle processes are described in ISO/IEC 12207.

6.3 EM2 : References

This clause shall provide references to normative and technical documents. If the evaluation module depends on the result of other evaluation modules, then these shall be mentioned here.

6.4 EM3 : Terms and definitions

This clause shall define technical terms used in the evaluation module. Alternatively, references to sources in which definitions can be found shall be made.

6.5 EM4 : Inputs and metrics

6.5.1 Input for the evaluation

This clause shall identify the inputs required for the evaluation. These shall be classified as product component, product information, supporting information and product in use information.

NOTE 1 Information classified as product component includes software requirements specification, software design description, program description, source code, executable code, and user documentation.

NOTE 2 Information classified as product information includes software requirements review report, software design review report, program review report, unit test report, and user documentation review report.

NOTE 3 Information classified as supporting information includes quality assurance plan, configuration management plan, program test plan, and description of programming language and compiler. Supporting information is not evaluated but only used as background information necessary for conducting the evaluation.

NOTE 4 Information classified as product in use information includes testing report and operation report describing the behavior of the system. System includes any associated hardware, software and users.

6.5.2 Data elements

This clause shall specify which data elements are to be extracted from the inputs.

NOTE 1 Examples of data elements are: Number of lines of code containing comments; frequency distribution of sentence length in the user-manual; number of words in each help message; number of failures observed per hour of operation; number of tokens of specified types found in each module by a specified lexical scan.

NOTE 2 In general, the data elements are the material from which measures are computed. But in some cases, the raw data may itself constitute metrics.

6.5.3 Metrics and measures

This clause shall describe how the measures are calculated from the data elements using metrics.

If the metrics are to be combined to obtain 'higher-order' metrics, then the dependency shall be made explicit.

All assumptions and preconditions to be satisfied before measuring shall be stated.

6.6 EM5 : Interpretation of results

6.6.1 Mapping of measures

This clause shall specify the meaning of the measures, that is, the interpretation of the results of the measurement. This includes the evaluation scale onto which the values obtained by the defined metrics are to be mapped. If the mapping is non-trivial then the details of the algorithms (functions) needed to make the mappings shall be defined or references shall be made to their sources. If several measures are obtained for a single characteristic, sub-characteristics or attribute, then the clause shall describe how they can be combined into a rating for the characteristic, sub-characteristics or attribute.

The accuracy of the measure shall be specified.

6.6.2 Reporting

This clause shall describe the contents of the report providing the result of applying the evaluation module. In some cases, visualization of obtained values is important and should be encouraged.

6.7 EMA : Application procedure

NOTE The inclusion of EMA is optional but, if included, it shall have the following contents.

6.7.1 Definition of Technical Terms Used

This clause shall define technical terms that are not defined in 6.4, but used in part EMA of the evaluation module, or referenced sources.

6.7.2 Resources Required

This clause shall specify the resources required when applying the evaluation module. This should include: Software tools required (any software tools required should be identified and both generic tool types and proprietary tools should be referenced); hardware/software needed; test-harness or other equipment; skills and qualifications (any special skills and qualifications (for example, certification) required by the evaluator or the evaluating organization should be identified); effort of application (the effort required for typical application of the evaluation module should be estimated and, if this effort depends on attributes of the product (for example, number of lines of code), an estimating algorithm should be given); and any other resources required.

6.7.3 Evaluation instructions

This clause shall describe full details of the procedure to be followed. This should include the selection of evidence (for example, sampling of code), the generation and recording of raw data, counting rules, algorithms for computing metrics from raw data, the recording of results, and requirements for the retention of working and final documentation. In particular, the steps to be taken to ensure traceability and repeatability of results should be highlighted. The described procedure should be compliant with ISO Guide 25.

6.7.4 Documentation

This clause shall outline the internal documentation resulting from the usage of the evaluation module. The outlined report should be compliant with ISO Guide 25.

Annex A
(informative)

Development of evaluation modules

This informative annex provides guidance for the process of developing new evaluation modules. The new ISO/IEC 9126 part 2, part 3 and part 4 may serve as input for this process. The evaluation module development process should include five steps:

A.1 Identification of the evaluation module requirements

When the need for a new evaluation module is identified and a decision is made to develop the evaluation module, the first step should be to identify the requirements for the new evaluation module. This includes identification of a quality model and quality characteristic or sub-characteristic. Also, the stringency of the evaluation should be decided.

A.2 Specification of the evaluation module

Based on the identified requirements for the evaluation module the next step is to specify the evaluation techniques and inputs to the evaluation (for example, source code) as well as the set of metrics and the fundamental set of data elements. Here ISO/IEC 9126 part 2, part 3 and part 4 may be useful.

A.3 Development of the evaluation procedure

This step takes the formal specification from the previous step and adds some procedural aspects. The interpretation of metrics and data elements should be explained in the context of the evaluation. The resources required should be estimated and detailed evaluation procedures should be developed. Trial application of the evaluation procedure may be necessary.

A.4 Description of the evaluation procedure

In this step the evaluation procedure developed in the previous step should be described according to the evaluation module format, that is, the description should conform to this part of ISO/IEC 14598.

A.5 Verification and validation of the evaluation module

The evaluation module should be reviewed (verified) against its specification. Experts in the area covered by the evaluation technique described should do this. The validation should ensure that the evaluation module represents state of the art technology and existing technology that is unfamiliar to the organization.

The evaluation module should be tested (validated) by different groups of people in different environments. The experience obtained should be fed back to the evaluation module development team as input for an update.

Annex B
(informative)

Example of an evaluation module – Fault density

Information technology - Software product evaluation - Evaluation Module: Fault density

B.0 Introduction

This evaluation module is used to determine the fault density of a program. Detecting a high number of faults during design and testing phase reduces potential faults in the software, which will cause failures in the operation phase. A high number of faults at the beginning of the operation phases cause frequent failures and decreases product reliability. Therefore it should be ensured that the fault density is below a specified threshold value before the program is put into operation.

In general, it is impossible to count the number of faults remaining in a software product. But by using the model and the historical data of fault detection, the number can be estimated. The fault density is calculated using this estimate. The general estimation procedure is as below:

1) Choose an appropriate Reliability Growth Model RGM to adopt, for example, an exponential RGM or S-shaped RGM.

2) Record the cumulative number of detected faults with time at a particular point in the test period.

3) Decide on the number of parameters of the RGM equation that are required to fit the curve to the set of data recorded.

4) As time (t) of the RGM equation tends to infinity, the estimated number of potential faults can be calculated.

B.1 Scope

B.1.1 Characteristics

Reliability – Maturity – Fault density

B.1.2 Level of evaluation

Level B as defined in ISO/IEC 14598-5.

B.1.3 Technique

A reliability growth modelling technique is used. The number of faults in the final software product is predicted using the reliability growth model.

B.1.4 Applicability

1) This is generally used during system testing and is applicable for all types of programming languages.

2) When it is necessary to compare the fault density value with another value for a program written in a different programming language, the size values should be normalized.

B.2 References

—

B.3 Terms and definitions

The following definitions apply to the evaluation module:

Fault
A defect in the software.

Failure
Any occurrence of an event (or non-occurrence of some events) usually from a set of defined events.

LOC (Lines of Code)
The number of lines of code.

ELOC (Erroneous Lines Of Code)
The number of lines of code, in which a failure is detected and modified.

EELOC (Estimated Erroneous Lines Of Code)
The estimated number of erroneous lines of code (ELOC).

NCLOC (Non-Commented Lines Of Code)
The number of lines of code without comments.

FDV (Fault Density Value)
The value that indicates the number of faults per unit of product volume.

B.4 Inputs and metrics

B.4.1 Input for the evaluation

The following sources are used as input to the evaluation:

1) Product Component : Source code

2) Product Information : Program test report, Program review report, Program verification report

B.4.2 Data elements

The following data elements must be collected in order to apply the evaluation technique:

1) The number of detected faults (ELOC).

2) Detected time of each failure. Time must be measured in a consistent way, for example, CPU time or calendar time.

3) The number of lines of code (NCLOC).

B.4.3 Metrics and measures

The fault density value is calculated from

FDV = (EELOC - ELOC) / NCLOC

B.5 Interpretation of results

B.5.1 Mapping of measures

FDV (No mapping of metrics in this case).

Experimental source of scale inside the company:

FDV<=10E-4	Excellent
10E-4<FDV<=10E-3	Good
10E-3<FDV<=10E-2	Fair
FDV>10E-2	Poor

NOTICE : The threshold values must be tailored in accordance with applied phase or software domain.

B.5.2 Reporting

The following information is reported as the result of applying the evaluation module:

1) Identification of source code

2) Fault density value: FDV

3) The corresponding rating {Excellent, Good, Fair, Poor}

B.6 Application procedure

B.6.1 Definition of technical terms used

RGM
Reliability Growth Model. The Reliability Growth Model used to estimate the number of erroneous lines of code.

B.6.2 Resources required

The following resources must be available when the evaluation module is applied:

Software Tools Required :

1) LOC counting tool

2) Modified line counting tool

3) Reliability data collection tool and analysis tool (optional)

Hardware/software Platform

No special requirements.

Test-harness or other Equipment

No special requirements, but use of a reliability data collection and analysis tool is recommended.

Skills and Qualification

Knowledge of applying the RGM is required.

Man-power Effort of Application

Most of the effort is related to testing and recording of failures. If a reliability and data collection tool is applied, the application of the RGM and calculating the FDV only requires limited effort.

Other Special Resources

No special requirements.

B.6.3 Evaluation instructions

1) Selection of Sample

Select sample source code. Sampling ratio should be more than half.

2) Generation of Raw Data

Failure data is extracted from test reports. If test reports are not available, testing is performed. To apply the reliability growth model RGM a minimum of XXX testing time is required.

For each failure, the location and time of occurrence is recorded.

Count modified ELOC with modification time and total NCLOC of each sample.

3) Algorithm

Estimate the number of potential erroneous LOC using the reliability growth model RGM. The estimated number of erroneous lines of code is found by applying the RGM model:

EELOC = RGM-({(Failure,Time)})

Calculate the fault density value:

FDV = (EELOC - ELOC) / NCLOC

Notice: How to use the RGM model or tool must be explained here in all details or alternatively references to a software tool and manual for making the estimation must be provided.

4) Requirements for retention of working and final document

Measure once a week, and analyse the trend during the testing phase.

Action: If the calculated value of fault density is larger than some specified value (rating=poor), the source code is reviewed, re-tested and debugged, and fault density is calculated again.

B.6.4 Documentation (Internal)

The following information is recorded for internal documentation:

1) Identification and version of source code sample

2) Identification and version of test documentation

3) The failure set used for estimating of EELOC

4) Date of application and responsible person

Annex C
(informative)

Example of an evaluation module – Functionality

Information technology - Software product evaluation - Evaluation module: Functionality

C.0 Introduction

This evaluation module is used to determine the “functionality” of a software system/component, where functionality is intended as the extent to which software component provides functions which meet stated and implied needs under specified conditions.

A correct evaluation is possible if there is a well-documented definition of the software/system requirements and if stated needs are well described. In every case, the evaluation will consider how it is documented.

ISO/IEC 9126 defines the “functionality” in terms of sub-characteristics, as described in following paragraphs. Each sub-characteristic is measurable through the measure of more sub-items.

The elementary items hereby described for each sub-characteristic will be an approximate level for the definition of the sub-characteristic value. A formula for this evaluation does not currently exist and the adoption of an emerging standard will give confidence that the work will find a common agreement in the context of his application.

The quality model relevant to this evaluation module considers more than one metric for each elementary item pertinent to each sub-characteristic. Metrics for each elementary item to be measured are described in “checklists”, with (y/n/d) possible answer for each question: y means yes, n means no and d means discard because not applicable.

The answer y/n to a question is related to the comparison of the metric value with an expected value: if the measured value is equal or better then the expected value then the answer is “yes”, otherwise the answer is “no”.

The “functionality” characteristic is to be evaluated with following rules:

$$Vc = \Sigma\, Vsc_i / nsc$$

$$|Vsc_j| = \Sigma\, m_i / (n\text{-}nd)$$

Vc is the measured value of the characteristic

Vsc_i is the measured value of the i-sub-characteristic

nsc is the number of sub-characteristics.

m_i is 1 if the i-answer is positive, otherwise is 0

n is the total number of the measures

nd is the number of discarded questions

C.1 Scope

C.1.1 Characteristics

Functionality metrics indicate if the software component satisfies the defined requirements. Also the software component shall satisfy implied user needs; in other words, implicit requirements for the typology of the software component under evaluation.

Metrics for functionality include indicators for 5 sub-characteristics:

— suitability

— accuracy

— interoperability

— compliance

— security.

Suitability metrics measure the ratio of the satisfied functions during testing and user operations to the required functions; in other words, functions performing tasks which do not conform to documented requirements.

Accuracy metrics measure the precision:

— the range of erroneous of calculation

— the difference between the actual and expected results of task performed

— inconsistency between actual operation procedure and documented procedure (for example, in manuals)

Interoperability metrics measure the communicability level of the software component with other systems, other software products other equipment:

— data transfer capability

— command exchange capability

Compliance metrics measure the standardisation level of the software component against regulation or rules of the environment.

Security metrics measure the defence performance level against illegal access and/or illegal operations.

C.1.2 Level of evaluation

ISO/IEC 14598-5 in annex B (informative) describes the criteria for the selection of the evaluation level.

The criticality of the characteristic measured in this module in respect to different aspects (security, economy,..) influences the definition of the accuracy of the evaluation, the definition of the quantity of measures to be executed, and the techniques to be used. The question that determines the evaluation level is: if the “functionality” does not meet the requirements, what kind of problem exists?

The following table indicates levels and conditions for each level, considering:

— safety aspects;

— economy aspects;

— security aspects;

— environmental aspects.

The choice of the level shall be made adopting at least the higher level resulting from the analysis of each aspect.

Levels	safety aspects	economy aspects	Security aspects	Environmental aspects
A	many people killed	financial disaster (company will not survive)	Protection of strategic data and services	unrecoverable environmental damage
B	threat to human lives	large economic loss (company engaged	Protection of critical data and services	recoverable environmental damage
C	damage to property, few people injured	Significant economic loss (company affected)	Protection against error risk	local pollution
D	small damage to property, no risk to people	Negligible economic loss	no specific risk identified	no environmental risk

C.1.3 Technique

The following table indicates the evaluation techniques adopted to evaluate the FUNCTIONALITY, starting from the row relative to the chosen level and going down; that is, if the evaluation level to be adopted is B, the techniques from level B to level D are applied for the evaluation.

Evaluation techniques for FUNCTIONALITY

Level A	formal proofs (adequate techniques does not currently exist for the evaluation the FUNCTIONALITY at level A)
Level B	component testing (white-box testing)
Level C	review, code inspection
Level D	functional testing (black-box testing)

The following is an introductory description of the possible techniques indicated in the table.

Formal proofs

The most common method of program proving is the method of “inductive assertions”. The goal is the development of a set of theorems about the software component under evaluation. The method begins by writing assertions about the software component’s input conditions and correct results. A separate proof is necessary to show that the program will always eventually terminate. Other proof techniques are the methods of “predicate transformers, “subgoal induction”, “computation induction”, “structural induction”, and “intermittent assertions”.

In the future, this evaluation module would be enlarged to cover this evaluation level, but presently this level for FUNCTIONALITY is not measurable.

Component testing

Each software component is tested against the requirements. Also, it is tested to the completely integrated software components.

White-box testing

This technique permits one to examine the internal structure of the software component: the tester derives test data from an examination of the program's logic.

White-box testing is concerned with the degree to which test cases exercise or cover the logic (source code) of the program. A useful coverage criterion is the "statement coverage"; that is to require every statement in the program to be executed at least once. Stronger logic-coverage criteria are "path coverage", "decision coverage", "condition coverage", "decision/condition coverage" and "multiple condition coverage".

Review

The process that involves the visual inspection/analysis of software components and all the pertinent documentation.

Code inspections

Code inspections involves the reading or visual inspection of a software component. The objective is to find errors, but not to find solutions to the errors. However, this technique is not able to find "high level" errors, such as in the design. Code inspection has to be considered as complementary to the computer-based inspection, such as static analysis of code.

Checklists

The review activity is based on a defined checklist that permits the activity to be repeated with few subjective criteria. Questions in the checklist must be as simple as possible; the goal of the question has to be elementary information. Checklists are subjected to revision, integration, and deletion based on the application's experience.

Static analysis of code

It is possible to deduce the existence (possible) of errors in software by analysing its source code. One type of static analysis is the "control flow", in which the source code is subdivided into segments and the relationship between the segments are examined to verify, for example, that there are no segments that cannot be executed or that there are no paths which do not reach a "STOP" statement.

Another type of analysis concerns the "call graph", or "structure of the software system", that describes the nesting of all the software units.

Functional testing

This is a process of attempting to find discrepancies between the software product and its external specification. The specification is analysed to derive a set of test cases. It is important to well define the test cases and apply specific techniques and methods (equivalence partitioning, boundary value, analysis, cause-effect graphing, error-guessing methods, ...).

Black-box testing

The software is viewed as a black box; that is, there is no concern about the internal behaviour and structure of the program. The tester is only interested in finding circumstances in which the program does not behave according to its specifications. If there is not adequate exhaustive testing, it will be necessary to define adequate test cases.

C.1.4 Applicability

The scope of this evaluation module is to determine the "functionality measure" for the software product under evaluation. This evaluation module is applicable when two kinds of conditions are present:

— condition concerning the requirement for the evaluation process:

Applicable when the specific requirements for the evaluation process for the characteristic under evaluation are satisfied (software requirements for Functionality).

— condition concerning the documentation input to the evaluation:

Applicable to the availability of the documentation input to the evaluation process.

Requirement for the Evaluation Process:

Software requirements for functionality

SUITABILITY

— All functional requirements shall be documented.

— Hardware architecture of the product shall be described in the documentation.

— Software architecture of the product shall be described in the documentation.

— All inputs, processing and outputs shall be defined.

— The software components given to the laboratory to be evaluated shall be well identified as software components under evaluation.

— All hw/sw requirements for testing shall be identified.

— The results of performed tests shall be documented.

— Test specifications shall be fully specified.

— All design components shall be traceable with respect to functional requirements.

— The hw/sw environment for each test shall be described.

— ACCURACY

— The programs and data shall be free from contradictions within themselves and with the documentation.

— All functions mentioned in the software documentation shall be completely and correctly executable.

SECURITY

— The security requirements shall be defined.

— All the security threats, security objectives and security enforcing functions shall be identified.

— The security enforcing functions shall fulfil security objectives.

NOTE This list of requirements need to be integrated with specific requirements for the software produce under evaluation.

Documentation INPUT to the Evaluation:

Applicable to the availability of the documentation input to the evaluation process

The following table describes the documents/components necessary for the evaluation process for each evaluation level. Also, the table indicates when in a generic software life cycle the evaluation module may be used.

In the table, all that is defined for a level is valid for the other lower levels.

	Software components, documentation required	Applicable Software Life Cycle Phase	Requirements for the evaluation
Level A	Software requirement review report, software requirement verification report, requirement measurement report, software design review report, software design verification report, design measurement report, user documentation review report, description of software specification methods and tools, description of design methods and tools, description of programming language and compilers	During all software life cycle phases	Not applicable
Level B	Program review report, program verification report, program review report, program measurement report, program test plan, program test report, unit test plan, unit test report, system requirement analysis, system specification and design, system test plan, configuration management plan, configuration management report, quality assurance plan, quality assurance report	During all software life cycle phases	Co-operation with the “developer” during the testing activities (7), availability to the “target environment” in which instrumented code is executed, availability of change information between “evaluation environment” and “target environment”.
Level C	Source code, description of programming language and compilers, software requirement specification(5), software design description, system review report, system verification report, system test plan, system test report	After the software development phase	source code written in C language(6), co-operation with the “developer” during the review process
Level D	Executable product (1), product description (2), user manual (3), system manuals, test cases (4)	Before the delivery of the product	

(1) This means either running the product in the evaluation environment or having the access to the target environment with the possibility of running the product there.

(2) Product description is a collection of information about what the user expects from the product. It may have the form of “product coversheet”, or “user requirements”, or other information.

(3) “User manual” is a collection of information about how to utilise the software product. It may also have the form of “interactive information”, with or without paper support.

(4) “Test cases” include test data and test results. The evaluation process will utilise such information, but is not limited to it.

(5) “Software requirement specification” means a collection of the essential requirements: functional requirements, performance design constrains of the software and its external interfaces.

(6) The restriction for the source code language is due to the availability for the laboratory of the “parser” (or precompiler, or interpreter) for the specified language. It is a temporary restriction, enlarged to other “languages” when the laboratory can dispose of the relative “parse”.

(7) “developer” in this context is the “organisation” that has produced the software product.

For detailed description about the documentation indicated in the table, refer to ISO/IEC 14598-5.

C.2 References

.....

C.3 Terms and definitions

.....

C.4 Inputs and metrics

C.4.1 Input for the evaluation

The following is the minimal input documentation for each evaluation level. See ISO/IEC 14598-5: Information technology - Software product evaluation - Part 5: Process for evaluators (*) for more detailed information about the contents of the documents.

The titles of the documents are “indicative”; they may change depending on the internal/standard documentation in the development environment. The contents required as described in (*) should be integrated/described in similar documents. It should be a useful across reference indicating where the information is needed for the evaluation.

Input for evaluation level A

Software requirement review report, software requirement verification report, requirement measurement report, software design review report, software design verification report, design measurement report, user documentation review report, description of software specification methods and tools, description of design methods and tools, description of programming language and compilers.

Program review report, program verification report, program review report, program measurement report, program test plan, program test report, unit test plan, unit test report, system requirement analysis, system specification and design, system test plan, configuration management plan, configuration management report, quality assurance plan, quality assurance report.

Source code, description of programming language and compilers, software requirement specification, software design description, system review report, system verification report, system test plan, system test report.

Executable product, product description, user manual, system manuals, test cases.

Input for evaluation level B

Program review report, program verification report, program review report, program measurement report, program test plan, program test report, unit test plan, unit test report, system requirement analysis, system specification and design, system test plan, configuration management plan, configuration management report, quality assurance plan, quality assurance report.

Source code, description of programming language and compilers, software requirement specification, software design description, system review report, system verification report, system test plan, system test report.

Executable product, product description, user manual, system manuals, test cases.

Input for evaluation level C

Source code, description of programming language and compilers, software requirement specification, software design description, system review report, system verification report, system test plan, system test report

Executable product, product description, user manual, system manuals, test cases.

Input for evaluation level D

Executable product, product description, user manual, system manuals, test cases.

C.4.2 Data elements

The information to be extracted from the input documents is two types:

— information to the “evaluation process”, for example, requirements list

— information useful to “understand the system”

The first type of information are data to the evaluation process; this kind of data is described in the following paragraph for each metric in which is utilised.

The second type of information is not subject to evaluation; it may consist of various informal documents furnished by the developer to the evaluator; it is part of the documentation of the evaluation (for example, fax, mails, etc.). These documents should be maintained by the evaluators with all other documentation (that is, working reports by measure activities).

This paragraph, 4.2 Data, is defined at a general level in this evaluation module. For evaluation modules at lower levels, that is, Evaluation module for a “metric” (Fault density), this paragraph should be more descriptive and precise.

C.4.3 Metrics and measures

The following is the decomposition of the characteristic FUNCTIONALITY in sub-characteristics and for each sub-characteristic the metric to be used for the evaluation is indicated.

Metrics for COMPLIANCE

Compliance with (project) software development standards ratio

The ratio of correctly applied rules related to project development standards to total number of software development standards rules.

Compliance with (project) documentation standards ratio

The ratio of correctly applied rules related to project documentation standards to total number of project documentation standards rules.

Standardised data format ratio

The ratio of standardised data formats to the data formats to be standardised.

Standardised character ratio

The ratio of standardised graphic characters and control characters to those to be standardised.

Metrics for SUITABILITY

Functions available ratio

The ratio of functions effectively at disposal to the user to total number of specified function.

Functional specification change ratio

The ratio of function that must be changed (change includes addition, modification and deletion) after entering operation (operation testing) to total number of specified function.

NOTE Specified function is function which is defined requirements specifications or which is provided by operable software or described in the user manual as user accessible functions.

Precision of Input-Output definition ratio

The ratio of input-output data clearly and correctly defined to total number of Input-Output.

Project documentation ratio

The ratio of project documents available with the product to total number of project documents required.

Product documentation ratio

The ratio of product documents available with the product to total number of product documents required.

Metrics for ACCURACY

Significant digits ratio

The ratio of the implemented significant digits to the required significant digits for the data items that require specific accuracy.

Volume of code ratio

The ratio of the actual volume of code to the required volume of code.

Correctness ratio

The ratio of obtained data with the needed degree of precision respect to the expected data.

Metrics for INTEROPERABILITY

Communicability ratio

The ratio of network communications that match to network communication standards.

The ratio of common technical vocabulary with the system and all inter-operating systems.

Matched data format ratio

The ratio of data formats that match to those of the other systems involved inter-operation.

Matched character ratio

The ratio of graphic characters and control characters matched to those of the other systems in interoperation.

Metrics for SECURITY

Software access control ratio

The ratio of unauthorised access to software to the number of attempts.

Data access control ratio

The ratio of unauthorised access/changes-to-data respect to the number of attempts.

Ciphered data ratio

The ratio of ciphered data to the data to be ciphered.

Access history ratio

The ratio of confidential information records that have access histories to all confidential information records. Access histories contain information on by whom, when and which of the confidential information records were accessed.

Data corruption

Frequency of data corruption during operation.

Detected abnormal operation ratio

The ratio of detected illegal operations to the input of illegal operations.

C.5 Interpretation of results

C.5.1 Mapping of measures

The evaluation scale for each sub-characteristic of FUNCTIONALITY is related to the percentage of positive answers to the preview questions (see Introduction). The following indicates the expected values for each sub-characteristic and the related rated value. The rated value for each sub-characteristic is input to the “formula” for the evaluation of the characteristic FUNCTIONALITY.

	Compliance	Suitability	Accuracy	Interoperability	Security
Expected values:	more than 25% positive answer	more than 70% positive answer	More than 70% positive answer	more than 70% positive answer	more than 70% positive answer

Rated value	Evaluated values for Compliance	Evaluated values for Suitability	Evaluated values for Accuracy	Evaluated values for Interoperability	Evaluated values for Security
1 (poor)	[0 .. 0.25]	[0 .. 0.70]	[0 .. 0.70]	[0 .. 0.70]	[0 .. 0.70]
2 (fair)	] 0.25 .. 0.50]	] 0.70 .. 0.80]	] 0.70 .. 0.80]	] 0.70 .. 0.80]	] 0.70 .. 0.80]
3 (good)	] 0.50 .. 0.75]	] 0.80 .. 0.00]	] 0.80 .. 0.00]	] 0.80 .. 0.00]	] 0.80 .. 0.00]
4 (excellent)	] 0.75 .. 1]	] 0.90 .. 1]	] 0.90 .. 1]	] 0.90 .. 1]	] 0.90 .. 1]

Starting from the sub-characteristic rated value, the following is the formula for the FUNCTIONALITY.

Value for FUNCTIONALITY:

$$Vc = \Sigma\, Vsc_i / nsc$$

0

where nsc is 5.

C.5.2 Reporting

The results of the evaluation shall be documented in an Evaluation Report with structure and contents compliant with ISO/IEC 14598-5 (Annex A).

The following summarises sections and contents for the required Evaluation Report template.

- **Section 1 - Identifications**

This section of the evaluation report contains identification information relative to the evaluation performed.

- **Identification of the evaluator**
  - — the name of the evaluator's organisation,
  - — the address of the evaluator's organisation,
  - — the location(s) where the evaluation has been carried out (if different from address above),
  - — the name of the person responsible for the evaluation.
- **Identification of evaluation report**
  - — unique identification of the report (for example, serial number),
  - — the number of pages in the report.

- **Identification of requester**

  — the name of requester's organisation,

  — the address of requester's organisation,

  — the name of software product supplier (if different from name above),

  — the address of software product supplier (if different from address above).

- **Section 2 - Evaluation requirements**

This section of the evaluation report shall contain the evaluation requirements:

  — a general description of the product application domain,

  — a general description of the product purpose,

  — the list of quality requirements and product information evaluated, possibly including reference to quality characteristics and evaluation levels.

- **Section 3 - Evaluation specification**

This section of the evaluation report shall contain the evaluation specification:

  — the scope of the evaluation, referring to the product description,

  — the cross-reference between the information requested in the evaluation requirements and the product components,

  — the specification of measurements and verifications,

  — the mapping between the specification of measurements and verifications and the evaluation requirements.

- **Section 4 - Evaluation methods**

This section shall contain the documentation of the evaluation methods used to perform the evaluation. For each evaluation method included here, the identification of product components on which the method has been applied shall be provided.

- **Section 5 - Evaluation Results**

This section of the evaluation report shall contain the evaluation results:

  — the evaluation results themselves,

  — intermediate results or interpretation decision, whenever necessary,

  — the reference to the tools used during the evaluation.

An Evaluation Report may include the results of more than one Evaluation Module, in that case the structure of the sections is:

- **Section 1 - Identifications**
- **Section 2 - Evaluation requirements**
- **Section 3 - Evaluation specification**
- **Section 4-A - Evaluation methods for Evaluation Module “XXXXX”**
- **Section 5-A - Evaluation Results for Evaluation Module “XXXXX”**
- **Section 4 B - Evaluation methods for Evaluation Module “YYYYYY”**
- **Section 5 B - Evaluation Results for Evaluation Module “YYYYY”**

……..

Annex D
(informative)

Example of an evaluation module – Usability and Quality in Use

Information technology - Software product evaluation - Evaluation module: Usability and Quality in Use

D.0 Foreword and introduction

Foreword

This evaluation module is for measuring quality in use as specified ISO 9126-1 and using the principles of ISO 9241-11. It is based on the MUSiC methodology [2].

Introduction

This evaluation module gives the principles for evaluating quality in use by evaluating the results of using the product, with a representative sample of users carrying out representative tasks in a simulated environment.

D.1 Scope

D.1.1 Characteristics

This evaluation module specifies how to evaluate the following three quality in use characteristics defined in ISO/IEC 9126-1:

Effectiveness: The capability of the software product to enable users to achieve specified goals with accuracy and completeness in a specified context of use.

Productivity: The capability of the software product to enable users to expend appropriate amounts of resources in relation to the effectiveness achieved in a specified context of use.

Satisfaction: The capability of the software product to satisfy users in a specified context of use.

NOTE ISO/IEC 9126-1 also defines safety as a characteristic, but safety is outside the scope of this evaluation module.

D.1.2 Level of evaluation

This evaluation procedure can provide accurate estimates of three characteristics of quality in use. The degree of accuracy depends on how closely the context of evaluation simulates the context of use, and on the number of users in each user group who are evaluated. At least eight users should be evaluated in a realistic context of use to obtain reliable results.

D.1.3 Technique

A sample of users who are representative of a particular user group attempt to achieve representative task goals using the product in a simulated environment without any assistance other than that available in the real working environment. Users also fill in a satisfaction questionnaire.

D.1.4 Applicability

The evaluation module is appropriate for any product which forms part of an system with which users interact to achieve task goals.

It can be applied during development, acquisition or operation, for quality assurance or validation. During development the evaluation module may be used to evaluate early prototypes using only 3 or 4 users, to obtain an indication of whether quality in use targets are likely to be met. During acquisition the evaluation module can provide assurance that the product is appropriate for the intended working environment. During operation, the evaluation module can establish baseline values against which to compare future products, and can indicate which product attributes may need to be improved.

D.2 References

[1] ISO 9241-11:1998, *Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability*.

[2] Macleod M, Bowden R, Bevan N and Curson I (1997) The MUSiC Performance Measurement method, Behaviour and Information Technology, 16.

[3] Brooke J (1996). SUS: A "quick and dirty" usability scale. In Usability Evaluation in industry. Taylor and Francis. See http://www.redhatch.co.uk/sus.html

[4] Lewis, J.R. (1995). IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use. International Journal of Human-Computer Interaction, 7, 57-78.

[5] Kirakowski, J. (1996). The software usability measurement inventory: background and usage. In: P. Jordan, B Thomas, & B Weerdmeester, Usability Evaluation in Industry. Taylor & Frances, UK. See also http://www.ucc.ie/hfrg/questionnaires/sumi/index.html

[6] Shneiderman, B. (1998). Designing the User Interface. Reading, MA, Addison-Wesley Publishing Co. See also: http://www.cs.umd.edu/projects/hcil/Research/1994/quis.html

D.3 Terms and definitions

Context of use: The users, tasks, equipment (hardware, software and materials), and the physical and social environments in which a product is used. [ISO 9241-11]

User: The person who interacts with the product. [ISO 9241-11]

Goal: An intended outcome. [ISO 9241-11]

Task: The activities required to achieve a goal. [ISO 9241-11]

D.4 Inputs and metrics

D.4.1 Input for the evaluation

D.4.1.1 Product component: working prototype

A working prototype is evaluated (including executable code and user documentation).

D.4.1.2 Product information: context of use

A definition of the intended contexts of use of the product is required, including the essential characteristics and capabilities of the intended user groups, their goals and tasks and the intended technical and support environment.

D.4.1.3 Supporting information: context of evaluation

The context of evaluation is a specification of the conditions under which the tasks are to be performed. It should be based on the intended context of use. The following information should be provided:

— The scenarios and task goals used in the evaluation.

— The configuration used for the evaluation including the hardware configuration, the operating system, and if the product is browser based, the browser used.

— The display devices if the product has a screen based visual interface, including the screen size and monitor resolution and the size and font used.

— The media size and print resolution, if the product has a print-based interface.

— The audio bits and volume setting, if the product has an audio interface.

— The manual input device (keyboard, mouse, joystick, etc.), if the product has a manual interface.

— The environment, setting or type of space in which the evaluation was conducted. For example: usability lab configured to simulate a cubicle office, meeting room, home office, home family room, manufacturing floor.

— Information about the evaluation participants: demographics, including age, gender and any special needs; how participants were selected and whether they have the same essential characteristics and capabilities as the intended users.

Any known differences between the context of evaluation and the intended context of use should be noted in the evaluation report.

D.4.2 Data elements

D.4.2.1 Task time

The total time taken by each user to complete the task (excluding any interruptions).

D.4.2.2 Task output

A concrete representation of the results of the task produced by each user (eg data, a paper record or the user's response to a questionnaire).

D.4.2.3 Satisfaction results

A completed satisfaction questionnaire.

NOTE Using a standardized measure provides data which is easier to interpret. Standardized questionnaires include SUS [3], PSSUQ [4], SUMI [5] and QUIS[6]).

D.4.2.4 Difficulties encountered

It is usually appropriate to provide additional qualitative data identifying the problems encountered by users which caused difficulties with quality in use. Suggestions may also be included for changes to the product which would improve quality in use.

D.4.3 Metrics and measures

D.4.3.1 Effectiveness

Effectiveness is a measure of whether users can achieve their goals accurately and completely. It does not take into account how the goals were achieved, only whether they were achieved.

Effectiveness should be measured by the extent to which the goals of the task have been achieved. One possible metric is the percentage of users who wholly achieve their goals. If goals can be partially achieved (e.g. by incomplete or sub-optimum results) then a more appropriate metric would be the average goal achievement, scored on a scale of 0 to 100% based on specified criteria. In some cases the percentage of users making uncorrected critical errors may be important.

D.4.3.2 Productivity

Productivity relates the level of effectiveness achieved to the quantity of resources expended. Efficiency is generally assessed by the mean time taken to achieve the task. Efficiency may also relate to other resources (e.g. total cost of usage), or be relatively unimportant (e.g. for some consumer applications).

Task time is the general measure of efficiency. It assumes that the less time a user takes to complete a task, the less resources the task takes, and the better the product is. Measuring efficiency as effectiveness/time gives a measure of work rate, and is useful when comparing different products for the same user group and task.

D.4.3.3 Satisfaction

Satisfaction is an assessment of the users' reaction to using the product and should be measured using a standardized questionnaire.

D.5 Interpretation of results

D.5.1 Mapping of measures

D.5.1.1 Effectiveness

Effectiveness is measured as a percentage. The criteria for effectiveness will depend on the evaluation level and the nature of the business objectives.

D.5.1.2 Productivity

Productivity can be measured by task time or effectiveness/task time. The criteria for efficiency will depend on the evaluation level and the nature of the business objectives.

D.5.1.3 Satisfaction

Criteria for satisfaction can be set by comparison with previous results for related products, or by comparison with a supplied database of industry standard norms for the questionnaire.

D.5.1.4 Interpretation of measures

D.5.1.5 Accuracy

All metrics reported should give mean values and the standard error of the mean. Any claims of differences between values should state the probability that the difference did not occur by chance.

D.5.1.6 Interpretation

Each measure should be interpreted in relation to the requirements for quality in use in a specific context of use. It is not generally meaningful to combine efficiency and satisfaction scores.

D.5.2 Reporting

A report should have the following contents:

— The purpose of the product: what the product is for, what it is intended to do for its users.

— The objectives of the evaluation, and any specific target values for the measures to be taken.

— The essential characteristics and capabilities that are expected of the user groups that are being evaluated.

— Context of evaluation: the conditions under which the tasks were performed, and any known differences between the evaluated context and the expected context of use.

— Evaluation design: the user groups, the tasks given, any other independent variables, and the measures taken.

— Procedure: sequence of events.

— Task Instructions.

— Results (which should include graphical representation).

— Difficulties encountered by users and suggestions for changes to the product (optional).

— Interpretation.

D.6 Application procedure

D.6.1 Resources Required

— An evaluator with skill or expertise in human factors evaluation.

— The minimum effort for the evaluators is approximately: 3 person days planning, 2 person days for the evaluation, and 2 person days for analysis and reporting.

— Use of a usability laboratory, or facilities to monitor the interaction remotely by video are desirable (but not essential).

D.6.2 Evaluation instructions

The purpose of the evaluation is to help readers of the report decide whether the product will have quality in use for their particular users, tasks and working environment. The design of the evaluation should be based on a simulation of the intended usage environment.

The evaluation and the results should be reported in sufficient detail to enable readers to be able to judge the relevance of the results to the needs of their own users, tasks and working environments.

The following guidelines will help ensure that the evaluation procedure is as close to real world usage as possible:

— The evaluation report needs to make it clear which users, tasks and working environments the product is intended for, and the extent to which these characteristics were actually simulated in the evaluation.

- The task instructions should tell users what they need to achieve, without giving any clues about which product features to use.

- To be representative of real world usage, the evaluation situation should be as natural as possible. This may mean simulating distracters and other working conditions. The evaluators should be as unobtrusive as possible (preferably observing remotely in another room).

  - During the evaluation participants should not be asked to think aloud.

  - Participants should not be given any hints or assistance, other than by mechanisms available to real users (such as documentation or a telephone help desk).

- Data should be obtained from sufficient users in each category for the sample of users to be representative of the intended user group. Given the typical variability of participants in an evaluation, it has been found that for consistent results it is best to evaluate at least 8 participants from each user group.

- 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.

Bibliography

[1] ISO/IEC Guide 2, *Standardization and related activities – General vocabulary*.

[2] ISO/IEC Guide 25, *General requirements for the competence of calibration and testing laboratories*.

[3] ISO/IEC TR 9126-2, *Software engineering — Product quality – Part 2: External metrics*.

[4] ISO/IEC TR 9126-3, *Software engineering — Product quality – Part 3: Internal metrics*.

ICS 35.080

Price based on 31 pages