# Usability Evaluation Based on International Standards for Software Quality Evaluation

KOMIYAMA Toshihiro

## Abstract

This paper is an introduction to the series of international software quality evaluation standards that have been established by ISO/IEC JTC1 SC7, which is the organization responsible for the international standardization of software engineering. One of the standards, "ISO/IEC 9126-1: Quality Model" models the structure of software quality and defines the quality characteristics and sub-characteristics that form the model. Usability is positioned as one of the six quality characteristics of the model. This paper focuses on usability and describes viewpoints on usability evaluation and the metrics of quantitative usability evaluation based on the international standards for software quality evaluation. In addition, it will also discuss the utilization of evaluation technologies that are aimed at improvements in usability.

Keywords

software quality evaluation, usability, quality model, quality characteristics, quality metrics

## 1. Introduction

Progress of the information society has been expanding the stage of computer use, and a huge amount of software is being developed and used. Following this trend, the effects of software defects on society and the lives of individuals have also increased, leading to a rise in interest into the quality of software. In addition, as this situation has also tended to spread the recognition that software exerts important effects on the values of systems use, interest in software quality is also being enhanced from the viewpoint of improvements in product values.

The improvement of software quality has been one of the major topics of discussion since the first international conference on software engineering was held under the sponsorship of NATO in 1968. The engineering approach to quality improvement includes the implementation and improvement of quality based on the utilization of design and verification techniques and tools. Additionally, from the management point of view, an explicit definition of quality goals and a technology for measuring, evaluating and controlling the achievement of such goals are also important.

The quality of software is not enough in itself if it is simply capable of providing the required functions (functionality) or if is free of defect or failure (reliability). In order to enhance user satisfaction and provide superiority in market competition with high quality, it is required to consider quality from multilateral viewpoints. These will include ease of use for the user (usability) and the speed between making a processing request and receiving the result (efficiency). An evaluation technology for each of the characteristics to be considered is also necessary.

When the user and developer make an agreement on quality requirements or evaluate products by comparing them, the discussions would be confused if their ways of recognizing and evaluating quality differed. From the awareness of this problem, ISO/IEC JTC1 SC7/WG6 [*1] (Evaluation and Metrics) is tackling international standardization of the structural model, quantification, evaluation technology and the evaluation process of software quality.

In the following sections, we will first introduce an overview of the ISO/IEC standards related to software quality evaluation, then explain about the usability evaluation tech-

[*1] JTC1 is a joint technical committee established in order to promote international standards for the information technology (IT) market in cooperation between ISO (The International Organization for Standardization) and IEC (The International Electro-technical Commission). SC7 is a sub committee of JTC1 that is responsible for the software engineering field. WG6 is a working group of SC7 that is responsible for software evaluation and metrics.

nology of these standards in the context of the human interface, features of this issue, and finally we will discuss points that are relevant to the use of these standards and their future perspectives.

## 2. International Standards for Software Quality Evaluation

### 2.1 History of Standardization

Based on the basic recognition that software quality should not be measured or evaluated only with regard to the number of defects or for using arbitrary selected measures, a proposal for standardization of software quality evaluation was made and the work for international standardization was started at ISO/TC97/SC7 in 1985. Since then, the project for standardization of software quality evaluation was taken over by the ISO-IEC joint committee, ISO/IEC JTC1, that was organized in 1987 and then assigned to WG6 in 1992, which was established in ISO/IEC JTC1 SC7 (Software Engineering).

In Japan, the Information Technology Research and Standardization Center of the Japanese Standards Association (JSA/INSTAC) established the Technical Committee on Software Quality Evaluation in 1987 and the Information Technology Standards Commission of Japan of the Information Processing Society of Japan (IPSJ/ITSCJ) established the WG6 in 1992. Through their technological and administrative contributions to this international standardization, Japan has been leading the international standardization of software quality evaluation. Since the start of ISO/IEC JTC1 SC7/WG6, Prof. Azuma, Waseda University, has assumed the Convener, Mr. Komiyama, NEC Corporation, the Secretariat and some Japanese members the project editors. Their technical contribution based on the domestic Technical Committee and WG activities includes proposals of action items for its international standardization, submission of draft standards, and comments on proposed draft standards.

At NEC, we have also contributed to this project in both its technological and administrative aspects by assuming the secretarial function jointly with IPSJ/ITSCJ and by compiling parts of the standard drafts based on evaluation technologies that have been developed in house, such as SQMAT (Software Quality Measurement and Assurance Technology).

### 2.2 Organization of International Standards

"ISO/IEC 9126: Information Technology - Software product evaluation - Quality characteristics and guidelines for their use" was issued in 1991 as the first international standard for software quality evaluation and it was translated into Japanese and issued as JIS X0129 in 1994.

These standards define six quality characteristics as viewpoints of software quality evaluation together with the basic evaluation processes. Later on, the two series of standards and two supplementary standards as shown in Table 1 were established to promote the application of standards in actual businesses and to improve the convenience of standard users, and their translation into JIS was also performed.

At present, the establishment of "ISO/IEC 25000 Series: Software product Quality Requirements and Evaluation (SQuaRE)", a series of next-generation software quality evaluation standards, is underway by reorganization and enhancing the existing standards (see **Fig. 1** ). SQuaRE features the following improvements:

1) Establishment of the process of software quality requirements definition using quality models and quality metrics;

2) Definition of the quality characteristics of data processed by software.

3) Definition of basic measures (e.g. volume of deliverables, number of faults) for use in the quantification calculations of quality measures (e.g. number of faults per unit of volume).

The following description will be based on the existing standards as shown in **Table 1** .

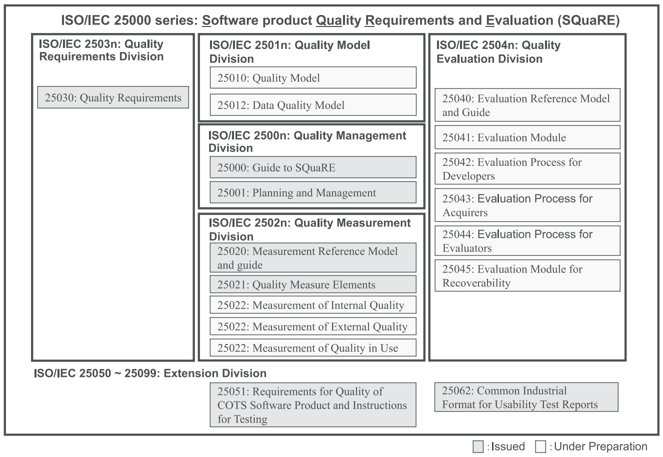

ISO/IEC 25000 series: Software product Quality Requirements and Evaluation (SQuaRE)
ISO/IEC 2503n: Quality Requirements Division
25030: Quality Requirements
ISO/IEC 2501n: Quality Model Division
25010: Quality Model
25012: Data Quality Model
ISO/IEC 2500n: Quality Management Division
25000: Guide to SQuaRE
25001: Planning and Management
ISO/IEC 2502n: Quality Measurement Division
25020: Measurement Reference Model and guide
25021: Quality Measure Elements
25022: Measurement of Internal Quality
25022: Measurement of External Quality
25022: Measurement of Quality in Use
ISO/IEC 2504n: Quality Evaluation Division
25040: Evaluation Reference Model and Guide
25041: Evaluation Module
25042: Evaluation Process for Developers
25043: Evaluation Process for Acquirers
25044: Evaluation Process for Evaluators
25045: Evaluation Module for Recoverability
ISO/IEC 25050 ~ 25099: Extension Division
25051: Requirements for Quality of COTS Software Product and Instructions for Testing
25062: Common Industrial Format for Usability Test Reports
▨ : Issued
□ : Under Preparation

Fig. 1 Architecture of SQuaRE standards.

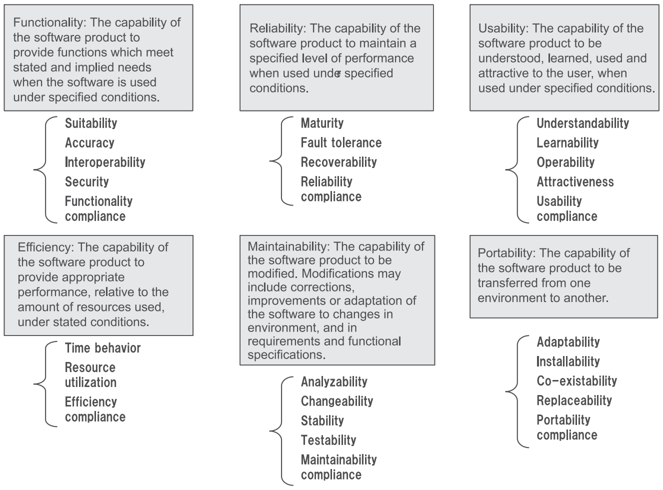
Table 1 International standards for software quality evaluation.

| Std. No.: Year<br>(JIS No.: Year) | Standard Name | Status | Outline |
|---|---|---|---|
| ISO/IEC 9126-1:2001<br>(JIS X 0129-1: 2003) | Software engineering - Product quality - Part 1: Quality model | Valid | Definition of quality characteristics and sub-characteristics |
| ISO/IEC TR 9126-2: 2003 | Software engineering - Product quality - Part 2: External metrics | Valid | Definition of external quality metrics |
| ISO/IEC TR 9126-3: 2003 | Software engineering - Product quality - Part 3: Internal metrics | Valid | Definition of internal quality metrics |
| ISO/IEC TR 9126-4: 2004 | Software engineering - Product quality - Part 4: Quality in use metrics | Valid | Definition of quality in use metrics |
| ISO/IEC 14598-1: 1999<br>(JIS X 0133-1: 1999) | Information technology - Software product evaluation - Part 1: General overview | Valid | Definition of basic terms for quality evaluation, basic concept |
| ISO/IEC 14598-2: 2000<br>(JIS X 0133-2: 2001) | Software engineering - Product evaluation - Part 2: Planning and management | Valid | Quality evaluation promotion planning, organizational support |
| ISO/IEC 14598-3: 2000<br>(JIS X 0133-3: 2001) | Software engineering - Product evaluation - Part 3: Process for developers | Valid | Quality evaluation process in the development phase |
| ISO/IEC 14598-4: 1999<br>(JIS X 0133-4: 2001) | Software engineering - Product evaluation - Part 4: Process for acquirers | Valid | Quality evaluation process in acquisition |
| ISO/IEC 14598-5: 1998<br>(JIS X 0133-5: 1999) | Information technology - Product evaluation - Part 5: Process for evaluators | Valid | Quality evaluation process by third parties |
| ISO/IEC 14598-6: 2001<br>(JIS X 0133-6: 2002) | Software engineering - Product evaluation - Part 6: Documentation of evaluation modules | Valid | Packaging of evaluation technology |
| ISO/IEC 12119: 1994<br>(JIS X 0152: 1995) | Information technology - Software packages - Quality requirements and testing | Valid | Definition of quality requirements for software packages and its testing |
| ISO/IEC 14756: 1999<br>(JIS X 0136: 2001) | Information technology - Measurement and rating of performance of computer-based software systems | Valid | Efficiency evaluation techniques |

## 3. Software Quality and Usability

### 3.1 Structure of Software Quality

ISO/IEC 9126-1 gives the structure of software quality and the definitions on quality characteristics. This standard divides software quality evaluation into the three phases of design to coding, testing and actual use, and recommends the use of evaluation viewpoints and metrics according to product features of each phase. The point is that the design to coding phase consists of static evaluation based on the results of a review of design documents and codes, the testing phase consists of dynamic evaluation based on the results of software operation on the computer system and the actual use phase consists of evaluation based on the effects of use of the system embedding the evaluation target software on the user. The evaluation viewpoints and metrics of these three phases are respectively referred to as the internal quality characteristics/metrics, the external quality characteristics/metrics and quality in use characteristics/metrics. However, common quality characteristics are defined for both the internal quality characteristics and the external quality characteristics because these are the evaluation viewpoints in the development phase and their viewpoints are shared although the metrics used in the evaluation are different.


Functionality: The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions.

Reliability: The capability of the software product to maintain a specified level of performance when used under specified conditions.

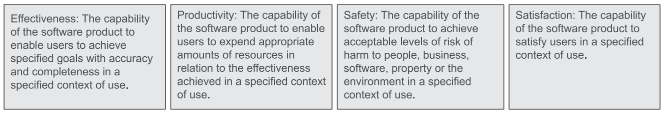
Usability: The capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions.

Efficiency: The capability of the software product to provide appropriate performance, relative to the amount of resources used, under stated conditions.

Maintainability: The capability of the software product to be modified. Modifications may include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications.

Portability: The capability of the software product to be transferred from one environment to another.

Fig. 2 Internal/external quality characteristics and sub-characteristics of 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.

Safety: The capability of the software product to achieve acceptable levels of risk of harm to people, business, software, property or the environment in a specified context of use.

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

Fig. 3 Quality in use characteristics of ISO/IEC 9126-1.

**Fig. 2** shows the internal and external quality characteristics and their lower-level characteristics (quality subcharacteristics) defined by ISO/IEC 9126-1, and **Fig. 3** shows the quality in use characteristics.

### 3.2 Quality Sub-Characteristics and Metrics of Usability

As shown in Fig. 2, ISO/IEC 9126-1 positions usability as one of the main quality characteristics to be considered in the development phase. Its quality sub-characteristics are defined as shown in **Table 2** , and the point of evaluation of each quality subcharacteristic is as follows.

**1) Understandability**

If the user can understand the functions and usage of software easily through familiar models, metaphors, etc.

**2) Learnability**

If the user can learn and acquire software skills without too

Table 2 The definitions of quality sub-characteristics of usability in ISO/IEC 9126-1.

| | | |
|---|---|---|
| Usability | Understandability | The capability of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use. |
| | Learnability | The capability of the software product to enable the user to learn its application. |
| | Operability | The capability of the software product to enable the user to operate and control it. |
| | Attractiveness | The capability of the software product to be attractive to the user. |
| | Usability compliance | The capability of the software product to adhere to standards, conventions, style guides or regulations relating to usability. |

much effort.

**3) Operability**

If the user can use the software without excessive stress, e.g. due to easy recovery from operational mistakes.

**4) Attractiveness**

If the display design, user interface, etc. can appeal to the senses of users.

**5) Usability Compliance**

If there is conformity with the standards and regulations related to human interface, such as the ISO 9241 series international standards, and with the style guidelines and rules determined by each organization or project.

In the definitions of the Quality in Use Characteristics of ISO/IEC 9126-1, satisfaction (see Fig. 3) is highly related to usability. It is obvious that taking usability into consideration from the development phase of design and implementation it will improve the satisfaction of users in the actual operation stage.

For the quality subcharacteristics of the usability and the satisfaction described above, ISO/IEC 9126-2, -3 and -4 define the metrics for their objective evaluation in each phase.

**Table 3** shows some examples of the usability metrics to be

Table 3 Examples of usability metrics of ISO/IEC 9126-2.

| Usability quality subcharacteristics | Metrics name | Purpose of the metrics | Method of application | Measurement, formula and data element computations | Interpretation of measured value |
|---|---|---|---|---|---|
| Understandability | Understandable input and output | Can users understand what is required as input data and what is provided as output by software system? | Conduct user test and interview user with questionnaires or observe user behaviour.<br>Count the number of input and output data items understood by the user and compare with the total number of them available for use | X= A / B<br>A= Number of input and output data items which user successfully understands<br>B= Number of input and output data items available from the interface | 0<=X<= 1<br>The closer to 1.0 is the better. |
| Learnability | Ease of function learning | How long does the user take to learn to use a function? | Conduct user test and observe user behaviour. | T= Mean time taken to learn to use a function correctly | 0<T<br>The shorter is the better. |
| Operability | Self-explanatory error messages | In what proportion of error conditions does the user propose the correct recovery action? | Conduct user test and observe user behaviour. | X= A / B<br>A =Number of error conditions for which the user proposes the correct recovery action<br>B =Number of error conditions tested | 0 <= X <= 1<br>The closer to 1.0 is the better. |
| Attractiveness | Interface appearance customisability | What proportion of interface elements can be customised in appearance to the user's satisfaction? | Conduct user test and observe user behaviour. | X= A / B<br>A= Number of interface elements customised in appearance to user's satisfaction<br>B= Number of interface elements that the user wishes to customise | 0 <= X <= 1<br>The closer to 1.0 is the better. |
| Usability compliance | Usability compliance | How completely does the software adhere to the standards, conventions, style guides or regulations relating to usability? | Specify required compliance items based on standards, conventions, style guides or regulations relating to usability.<br>Design test cases in accordance with compliance items.<br>Conduct functional testing for these test cases. | X = 1 - A / B<br>A= Number of usability compliance items specified that have not been implemented during testing<br>B= Total number of usability compliance items specified | 0<= X <=1<br>The closer to 1.0 is the better. |

used in the testing phase defined in ISO/IEC 9126-2. The evaluation of satisfaction is generally performed after the actual operation has advanced by a certain degree, by sending questionnaires to users selected by sampling.

## 4. Utilization of Software Quality Evaluation Standards

ISO/IEC 14598-1 defines the process of software quality evaluation as shown in **Fig. 4** . In the following, we will discuss points of evaluation in practice using the series of software quality evaluation standards by following the process flow.

### (1) Establish Evaluation Requirements

**1) Establish the Purpose of Evaluation**

Clarify the purpose of software evaluation by considering who wants what kind of evaluation and in which phase. For example, a project manager in the design phase may want to know the compliance to the screen design guide according to the customer requirements. Alternatively, a product planner may want to know the user satisfaction after shipment in order to prepare for enhancement of the next product.

**2) Identify Types of Product(s) to be Evaluated**

Based on the purposes of evaluations of multiple stakeholders, specify what is evaluated and at which stage of the software lifecycle. Usually, the evaluation targets in the design and implementation phases are the specifications and source codes, those in the testing phase are the results of the operation of the executable software, and those in the operation and maintenance phases are the effects exerted by the software on the users.

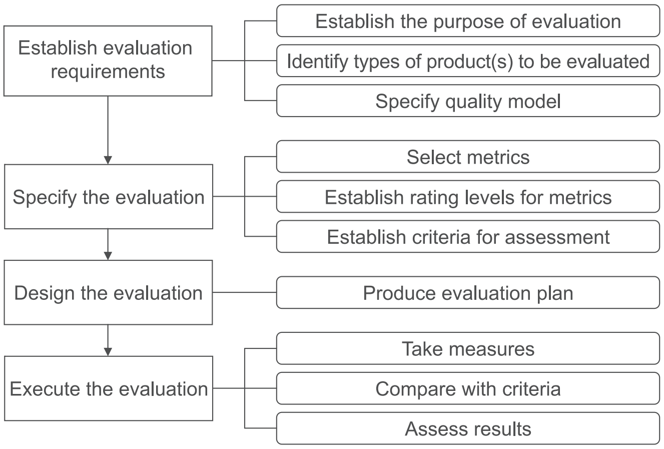

Establish evaluation requirements
Establish the purpose of evaluation
Identify types of product(s) to be evaluated
Specify quality model
Specify the evaluation
Select metrics
Establish rating levels for metrics
Establish criteria for assessment
Design the evaluation
Produce evaluation plan
Execute the evaluation
Take measures
Compare with criteria
Assess results

Fig. 4 Software quality evaluation process of ISO/IEC 14598-1.

**3) Specify Quality Model**

Clarify the quality characteristics and subcharacteristics required for the software product based on the quality models in ISO/IEC 9126-1, etc. This task includes the selection of characteristics to be considered, their prioritization and addition of missing characteristics. In addition, regard to usability is necessary to determine the necessity of usability evaluation and prioritization of subcharacteristics in order to implement and evaluate quality according to requirements.

### (2) Specify the Evaluation

**1) Select Metrics**

Clarify the quality metrics to be used for evaluating each characteristic in each phase. These should be selected from the metrics listed in ISO/IEC TR 9126-2, -3 and -4, revised considering the properties of the software or project, and defined in detail for the practical levels. If selection from ISO/IEC TR 9126-2, -3 and -4 metrics is not enough, it is also possible to define and add the original metrics.

**2) Establish Rating Levels for Metrics**

Define the rating levels for each metric to decide degree of satisfaction of the obtained values. This is the clarification of the acceptable range of values and the limits beyond which the product quality is unacceptable. The value should be as large as possible, as small as possible or as close as possible to a certain value depending on the metrics.

**3) Establish Criteria for Assessment**

In each phase where Go-NoGo judgment is required, for example decision of phase transition or product release, clarify the assessment criteria based on the quality data obtained using various metrics as well as the data on progress or cost, if necessary. The criteria may for example be set based on the overall score obtained with weighted averaging or using a decision table.

### (3) Design the Evaluation

**1) Produce Evaluation Plan**

Plan for quality evaluation, including when and what kind of data is collected by whom and also when the data is aggregated, evaluated and assessed according to the metrics and criteria defined above.

### (4) Execute the Evaluation

**1) Take Measures**

Collect data according to the plan and metrics defined above and calculate the measurement values of the metrics.

**2) Compare with Criteria**
Evaluate the calculated measurement value of each metric with respect to the criterion.

**3) Assess Results**
Evaluate the series of measurement values based on the assessment criteria and perform management judgments.

## 5. Present Status and Future Perspectives of Usability Evaluation

In general, each organization or project implements and evaluates usability by using the guidelines of the user interface design, reviewing based on usability checklists and testing carried out by the QA department, taking user requirements, product properties and user properties into consideration. Usability is an important factor for promoting satisfaction among software users and for differentiating a specific software product from a competing one. It is thus necessary for usability to adopt a systematic approach to evaluation and improvement throughout the lifecycle of a software product by fully utilizing the software quality evaluation standards introduced in this paper. What is important is to clarify the usability requirements by considering the related quality characteristics from the requirement definition and product planning stages. Furthermore the implementation of such requirements should be ensured by conducting quantitative, objective evaluations based on quality metrics in each phase of the development as well as after the start of actual operation.

In the future, we will continue to promote the development of more attractive software for users by accumulating usability evaluation data and expertise, setting the scales for deciding degree of satisfaction of each usability metrics, analyzing the relationship between the internal/external usability metrics and user satisfaction and so on.

### References

1) ISO/IEC 9126-1: Software engineering -- Product quality -- Part 1: Quality model, 2001 (JIS X 0129-1: 2003)
2) ISO/IEC 9126-2: Software engineering -- Product quality -- Part 2: External metrics, 2003
3) ISO/IEC 9126-3: Software engineering -- Product quality -- Part 3: Internal metrics, 2003
4) ISO/IEC 9126-4: Software engineering -- Product quality -- Part 4: Quality in use metrics, 2004
5) ISO/IEC 14598-1: Information technology -- Software product evaluation -- Part 1: General overview, 1999 (JIS X 0133-1: 1999)
6) ISO/IEC 25000: Software Engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE, 2005
7) AZUMA, M. et al: Software Quality Evaluation Guidebook (in Japanese), Japanese Standards Association, 1994.

### Author's Profile

**KOMIYAMA Toshihiro**
Chief Manager,
Software Engineering Division,
Software Business Promotion Unit,
NEC Corporation