

INTERNATIONAL STANDARD

ISO/IEC
9126

First edition
1991-12-15

Information technology — Software product evaluation — Quality characteristics and guidelines for their use

*Technologies de l'information — Évaluation des produits logiciels —
Caractéristiques de qualité de logiciel et directives d'utilisation*



Reference number
ISO/IEC 9126:1991(E)

Contents

1	Scope	1
2	Normative references	1
3	Definitions	1
4	Software quality characteristics	2
4.1	Functionality	2
4.2	Reliability	3
4.3	Usability	3
4.4	Efficiency	3
4.5	Maintainability	3
4.6	Portability	3
5	Guidelines for use of quality characteristics	4
5.1	Usage	4
5.2	Views of software quality	4
5.3	Evaluation process model	6
 Annexes		
A	Quality subcharacteristics	9
B	History of the work	12

© ISO/IEC 1991

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case Postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Foreword

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

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 9126 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

Annexes A and B of this International Standard are for information only.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9126:1991

Introduction

As software applications have grown, so too has the importance of software quality. In order to manage software quality, the technology to specify and evaluate both the software product quality and development process quality objectively and quantitatively is most important. However, a framework is necessary that provides for the evaluation of software quality. This International Standard was produced as one part of a set of documents which provides that framework.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9126:1991

Information technology — Software product evaluation — Quality characteristics and guidelines for their use

1 Scope

This International Standard defines six characteristics that describe, with minimal overlap, software quality. These characteristics provide a baseline for further refinement and description of software quality. Guidelines describe the use of quality characteristics for the evaluation of software quality.

This International Standard does not provide subcharacteristics and metrics, and method for measurement, rating, and assessment. This International Standard adheres to the definition of quality in ISO 8402.

NOTE - A proposal for definitions of subcharacteristics is provided for information in annex A.

The definition of the characteristics and associated quality evaluation process model in this International Standard are applicable when specifying the requirements for and evaluating the quality of software products throughout their life cycle.

Its characteristics may be applicable to every kind of software, including computer programs and data contained in firmware.

This International Standard is intended for those associated with acquisition, development, use, support, maintenance, or audit of software.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 2382-20 :1990, *Information technology -- Vocabulary - Part 20 : Systems development*.
ISO 8402 : 1986, *Quality - Vocabulary*.

3 Definitions

For the purpose of this International Standard, the following definitions apply.

3.1 assessment : An action of applying specific documented assessment criteria to a specific software module, package, or product for the purpose of determining acceptance or release of the software module, package or product.

3.2 features : Features are identified properties of a software product which can be related to the quality characteristics.

NOTE - Examples of features include path length, modularity, program structure, and comments.

ISO/IEC 9126: 1991(E)

3.3 firmware : Hardware that contains a computer program and data that cannot be changed in its user environment. The computer program and data contained in firmware are classified as software; the circuitry containing the computer program and data is classified as hardware.

3.4 level of performance : The degree to which the needs are satisfied, represented by a specific set of values for the quality characteristics.

3.5 measurement : The action of applying a software quality metric to a specific software product.

3.6 quality : The totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs (ISO 8402:1986).

NOTE - In a contractual environment, needs are specified, whereas in other environments, implied needs should be identified and defined (ISO 8402 : 1986, note 1).

3.7 rating : The action of mapping the measured value to the appropriate rating level. Used to determine the rating level associated with the software for a specific quality characteristic .

3.8 rating level : A range of values on a scale to allow software to be classified (rated) in accordance with the stated or implied needs. Appropriate rating levels may be associated with the different views of quality i.e. Users, Managers or Developers. These levels are called rating levels.

NOTE - These rating levels are different from the "grades" defined in ISO 8402.

3.9 software : Programs, procedures, rules and any associated documentation pertaining to the operation of a computer system.

3.10 Software product : A software entity designated for delivery to a user.

3.11 software quality : The totality of features and characteristics of a software product that bear on its ability to satisfy stated or implied needs.

3.12 software quality assessment criteria : The set of defined and documented rules and conditions which are used to decide whether the total quality of a specific software product is acceptable or not. The quality is represented by the set of rated levels associated with the software product.

3.13 software quality characteristics : A set of attributes of a software product by which its quality is described and evaluated. A software quality characteristic may be refined into multiple levels of sub-characteristics.

3.14 software quality metric : A quantitative scale and method which can be used to determine the value a feature takes for a specific software product.

4 Software quality characteristics

Software quality may be evaluated by the following characteristics.

4.1 Functionality

A set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs.

NOTES

1 This set of attributes characterizes what the software does to fulfil needs, whereas the other sets mainly characterize when and how it does.

2 For the stated and implied needs in this characteristic, the note to the definition of quality applies, (see 3.6).

4.2 Reliability

A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time.

NOTES

1 Wear or ageing does not occur in software. Limitations in reliability are due to faults in requirements, design, and implementation. Failures due to these faults depend on the way the software product is used and the program options selected rather than on elapsed time.

2 In the definition of ISO 8402, reliability is "The ability of an item to perform a required function...". In this document, functionality is only one of the characteristics of software quality. Therefore, the definition of reliability has been broadened to "maintain its level of performance..." instead of "...perform a required function" (see also 3.4).

4.3 Usability

A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.

NOTES

1 "Users" may be interpreted as most directly meaning the users of interactive software. Users may include operators, end users and indirect users who are under the influence or dependent on the use of the software. Usability must address all of the different user environments that the software may affect, which may include preparation for usage and evaluation of results.

2 Usability defined in this International Standard as a specific set of attributes of software product differs from the definition from an ergonomic point of view, where other characteristics such as efficiency and effectiveness are also seen as constituents of usability.

4.4 Efficiency

A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions.

NOTE - Resources may include other software products, hardware facilities, materials, (e.g. print paper, floppy disks) and services of operating, maintaining, or sustaining staff.

4.5 Maintainability

A set of attributes that bear on the effort needed to make specified modifications.

NOTE - Modification may include corrections, improvements or adaptation of software to changes in environment, and in requirements and functional specifications.

4.6 Portability

A set of attributes that bear on the ability of software to be transferred from one environment to another.

NOTE - The environment may include organizational, hardware or software environment.

5 Guidelines for use of quality characteristics

5.1 Usage

This International Standard is applicable to defining software quality requirements and evaluating (measuring, rating and assessing) software products including

- Defining the software product quality requirement.
- Evaluating software specification to see if it will satisfy the quality requirement during development.
- Describing features and attributes of the implemented software, (e.g. in users' manuals).
- Evaluating developed software before delivery.
- Evaluating the software before acceptance

Currently, only a few generally accepted metrics exist for the characteristics described in this International Standard. Standards groups or organizations may establish their own evaluation process models and methods for creating and validating metrics associated with these characteristics to cover different areas of application and lifecycle stages. In those cases where appropriate metrics are unavailable and cannot be developed, verbal descriptions or "rule of thumb" may sometimes be used.

To use the six quality characteristics for definition and evaluation purposes it is also necessary to establish rating levels and criteria specific to the organization or the application or both.

Metrics, rating levels, and criteria applied for quality evaluation should be stated when the evaluation results are communicated.

Though there is no widely accepted software classification system there are some classes of software that are widely accepted. The importance of each quality characteristic varies depending on the class of the software. For example, reliability is most important for a mission critical system software, efficiency is most important for a time critical real time system software, and usability is most important for an interactive end user software.

The importance of each quality characteristic also varies depending on the view points considered.

5.2 Views of software quality

There are several views of quality, some of which are discussed below.

5.2.1 Users' view

The definition of quality in ISO 8402 reflects the users' view, as do the characteristics defined in this International Standard.

Users are mainly interested in using the software, its performance and the effects of using the software. Users evaluate the software without knowing the internal aspects of the software, or how the software is developed.

Users' questions may include

- Are the required functions available in the software?
- How reliable is the software?
- How efficient is the software?
- Is the software easy to use?
- How easy is it to transfer the software to another environment?

5.2.2 Developers' view

The process of development requires the user and the developer to use the same software quality characteristics, since they apply to requirement and acceptance. When developing off-the-shelf software, the implied needs must be reflected in the quality requirement.

Since developers are responsible for producing software which will satisfy quality requirements they are interested in the intermediate product quality as well as in the final product quality. In order to evaluate the intermediate product quality at each phase of the development cycle, the developers have to use different metrics for the same characteristics because the same metrics are not applicable to all phases of the life cycle. For example, the user thinks of efficiency in terms of response time, while the developer has to use terms of path-length and access and waiting time in the design specification. Generally speaking, metrics applying to the external interface of a product are replaced by those applying to its structure.

The developers' view must also incorporate the view of the quality characteristics required by those maintaining the software.

5.2.3 Manager's view

A manager may be more interested in the overall quality rather than in a specific quality characteristic, and for this reason will need to assign weights, reflecting business requirements, to the individual characteristics.

The manager may also need to balance the quality improvement with management criteria such as schedule delay or cost overrun, because he wishes to optimize quality within limited cost, human resources and time-frame.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9126:1991

5.3 Evaluation process model

Figure 1 shows the major steps required to evaluate software quality starting with the quality characteristics defined in this International Standard. Due to the high-level nature of figure 1, a number of detailed procedures such as analysis and validation of metrics are not shown.

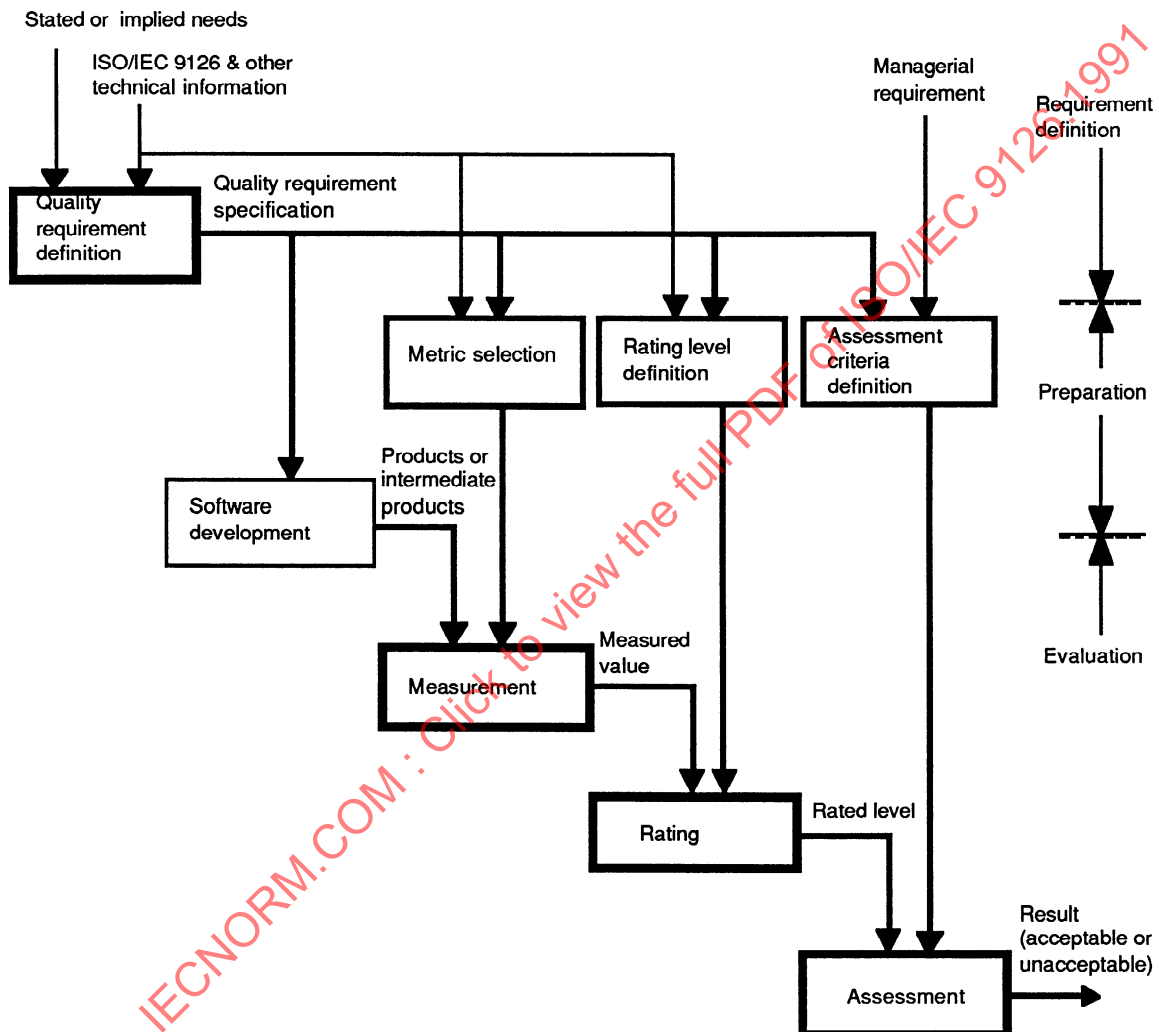


Figure 1 - Evaluation process model

The process consists of three stages: Quality Requirement Definition, Evaluation Preparation and Evaluation Procedure. This process may be applied in every appropriate phase of the life cycle for each component of the software product.

5.3.1 Quality requirement definition

The purpose of the initial stage is to specify requirements in terms of quality characteristics and possible subcharacteristics. Requirements express the demand of the environment for the software product under consideration, and must be defined prior to the development. As a software product is decomposed into major components, the requirements derived from the overall product may differ for the different components.

5.3.2 Evaluation preparation

The purpose of the second stage is to prepare the basis for evaluation.

5.3.2.1 Quality metrics selection

The manner in which quality characteristics have been defined does not allow their direct measurement. The need exists to establish metrics that correlate to the characteristics of the software product. Every quantifiable feature of software and every quantifiable interaction of software with its environment that correlates with a characteristic can be established as a metric.

Metrics can differ depending on the environment and the phases of the development process in which they are used. Metrics used in the development process should be correlated to the user respective metrics, because the metrics from the user's view is crucial.

5.3.2.2 Rating levels definition

Quantifiable features can be measured quantitatively using quality metrics. The result, i.e. measured value, is mapped on the scale. This value does not show the level of satisfaction. For this purpose these scales must be divided into ranges corresponding to the different degrees of satisfaction of the requirements (see figure 2). Since quality refers to given needs, no general levels for rating are possible. They must be defined for each specific evaluation.

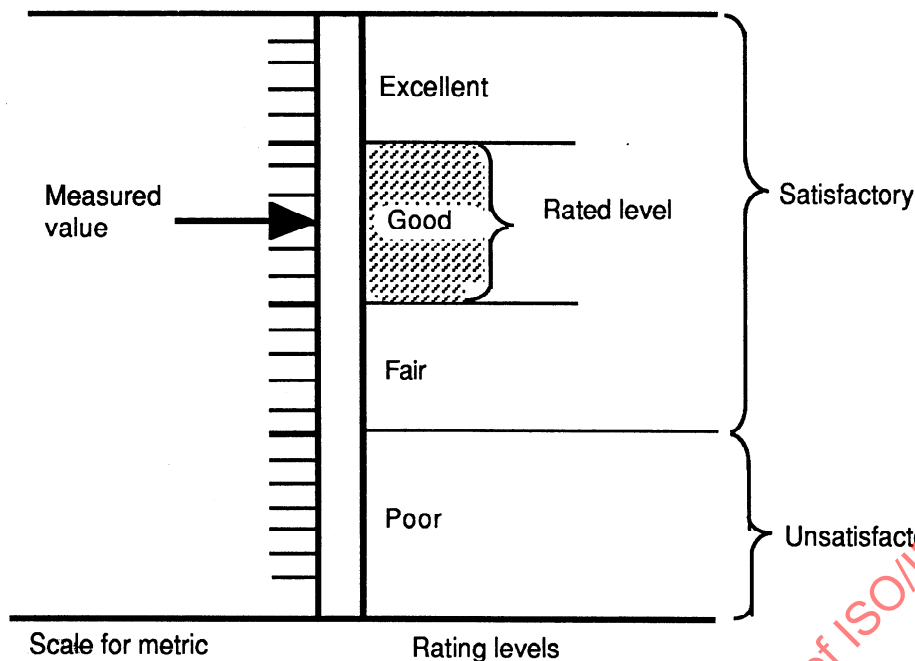


Figure 2 - Measured value and rated level

5.3.2.3 Assessment criteria definition

To assess the quality of the product, the results of the evaluation of the different characteristics must be summarized. The evaluator has to prepare a procedure for this, using, for instance, decision tables or weighted averages. The procedure usually will include other aspects such as time and cost that contribute to the assessment of quality of a software product in a particular environment.

5.3.3 Evaluation procedure

The last stage of the Evaluation Process Model is refined into three steps, namely measurement, rating and assessment.

5.3.3.1 Measurement

For measurement, the selected metrics are applied to the software product. The result are values on the scales of the metrics.

5.3.3.2 Rating

In the rating step, the rating level is determined for a measured value (see figure 2).

5.3.3.3 Assessment

Assessment is the final step of the software evaluation process where a set of rated levels are summarized. The result is a statement of the quality of the software product. Then the summarized quality is compared with the other aspects such as time and cost. Finally managerial decision will be made based on the managerial criteria. The result is a managerial decision on the acceptance or rejection, or on the release or no-release of the software product.

Annex A (informative) Quality subcharacteristics

A.1 Introduction

This annex provides an illustrative quality model which defines the characteristics of ISO 9126 into terms of subcharacteristics. This is a necessary step towards quality measurement using the quality evaluation process model described in ISO 9126. Subsequent related documents will deal with the measurement of subcharacteristics.

There are a number of such quality models in the literature and applied in practice. The maturity of the models, terms and definitions does not yet allow them to be included in a standard. Nevertheless, they are published to encourage use in practice and to collect experience for further editions. The key point is that there should be a quality model to at least the subcharacteristic level for a software product, not that it should be of the precise form described in this annex.

A.2 Definition of quality subcharacteristics

A.2.1 Functionality

A.2.1.1 Suitability

Attribute of software that bears on the presence and appropriateness of a set of functions for specified tasks.

NOTE - Examples of appropriateness are task oriented composition of functions from constituent sub-functions, capacities of tables.

A.2.1.2 Accuracy

Attributes of software that bear on the provision of right or agreed results or effects.

NOTE - For example, this includes the needed degree of precision of calculated values.

A.2.1.3 Interoperability

Attributes of software that bear on its ability to interact with specified systems.

NOTE - Interoperability is used in place of compatibility in order to avoid possible ambiguity with replaceability (see A.2.6.4).

A.2.1.4 Compliance

Attributes of software that make the software adhere to application related standards or conventions or regulations in laws and similar prescriptions.

A.2.1.5 Security

Attributes of software that bear on its ability to prevent unauthorized access, whether accidental or deliberate, to programs and data.

A.2.2 Reliability

A.2.2.1 Maturity

Attributes of software that bear on the frequency of failure by faults in the software.

ISO/IEC 9126: 1991 (E)

A.2.2.2 Fault tolerance

Attributes of software that bear on its ability to maintain a specified level of performance in cases of software faults or of infringement of its specified interface.

NOTE - The specified level of performance includes fail safe capability.

A.2.2.3 Recoverability

Attributes of software that bear on the capability to re-establish its level of performance and recover the data directly affected in case of a failure and on the time and effort needed for it.

A.2.3 Usability

A.2.3.1 Understandability

Attributes of software that bear on the users' effort for recognizing the logical concept and its applicability.

A.2.3.2 Learnability

Attributes of software that bear on the users' effort for learning its application (for example, operation control, input, output).

A.2.3.3 Operability

Attributes of software that bear on the users' effort for operation and operation control.

A.2.4 Efficiency

A.2.4.1 Time behaviour

Attributes of software that bear on response and processing times and on throughput rates in performing its function.

A.2.4.2 Resource behaviour

Attributes of software that bear on the amount of resources used and the duration of such use in performing its function.

A.2.5 Maintainability

A.2.5.1 Analysability

Attributes of software that bear on the effort needed for diagnosis of deficiencies or causes of failures, or for identification of parts to be modified.

A.2.5.2 Changeability

Attributes of software that bear on the effort needed for modification, fault removal or for environmental change.

A.2.5.3 Stability

Attributes of software that bear on the risk of unexpected effect of modifications.

A.2.5.4 Testability

Attributes of software that bear on the effort needed for validating the modified software.

NOTE - Values of this subcharacteristic may be altered by the modifications under consideration.

A.2.6 Portability

A.2.6.1 Adaptability

Attributes of software that bear on the opportunity for its adaptation to different specified environments without applying other actions or means than those provided for this purpose for the software considered.

A.2.6.2 Installability

Attributes of software that bear on the effort needed to install the software in a specified environment.

A.2.6.3 Conformance

Attributes of software that make the software adhere to standards or conventions relating to portability.

A.2.6.4 Replaceability

Attributes of software that bear on the opportunity and effort of using it in the place of specified other software in the environment of that software.

NOTES

1 Replaceability is used in place of compatibility in order to avoid possible ambiguity with interoperability (see A.2.1.3).

2 Replaceability with a specific software does not imply that this software is replaceable with the software under consideration.

3 Replaceability may include attributes of both installability and adaptability. The concept has been introduced as a subcharacteristic of its own because of its importance.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9126:1991

Annex B (informative) History of the work

B.1 Background

The software industry is entering a period of some maturing, while at the same time software is becoming a crucial component of many of today's products. This pervasive aspect of software makes it a major new factor in trade. Furthermore, with new global demands for safety and quality, the need for international agreements on software quality assessment procedures is becoming important.

There are essentially two approaches that can be followed to ensure product quality, one being assurance of the process by which the product is developed, and the other being the evaluation of the quality of the end product. Both avenues are important and both require the presence of a system for managing quality. Such a system identifies the management commitment to quality, and states its policies, as well as the detailed steps that must be in place.

To evaluate the quality of a product through some quantitative means, a set of quality characteristics that describe the product and form the basis for the evaluation is required. This International Standard defines these quality characteristics for software products.

B.2 History

State of the art in software technology does not yet present a well established and widely accepted description scheme for assessing the quality of software product. Much work has been done since about 1976 by a number of individuals to define a software quality framework. Models by McCall, Boehm, the US Air Force, and others have been adopted and enhanced over the years. However, today it is difficult for a user or consumer of software products to understand or compare the quality of software.

For a long time, reliability has been the only way to gauge quality. Other quality models have been proposed and submitted for use. While studies were useful, they also caused confusion because of the many quality aspects offered. Thus, the need for one standard model came about.

It is for this reason that the ISO/IEC technical committee began work to develop the required consensus and encourage standardization on a worldwide basis.

First considerations originated in 1978, and in 1985 the development of this International Standard was started. The models proposed initially introduced properties of software that depend on application or implementation aspects (or both), to describe the quality of software.

The first step of the ISO technical committee to arrange these properties systematically failed for lack of definitions. Terms were interpreted in different ways by different experts. All structures discussed were, therefore, of an arbitrary nature, without a common basis.

As a result it was decided that the best chance for establishing an International Standard was to stipulate a set of characteristics based on the definition of quality that later became part of ISO 8402. This international definition is accepted for all kinds of products and services. It starts with the user's needs.