ISO 9126: Analysis of Quality Models and Measures

Chapter · September 2010

DOI: 10.1002/9780470606834.ch10

2 authors:


Alain Abran
École de Technologie Supérieure


Rafa E. Al-Qutaish

SOFTWARE METRICS AND SOFTWARE METROLOGY

ALAIN ABRAN

WILEY

IEEE

IEEE computer society

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data is available.

ISBN: 978-0-470-59720-0

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

CONTENTS

ACKNOWLEDGMENTS

A number of collaborators, including colleagues in industry and university as well as PhD students, have helped me over the years improve my understanding of many of the concepts presented in this book, in particular:

Chapter	Co-Contributor
2: From Measurement Methods to Quantitative Models: A Measurement Context Model	Dr. Jean-Philippe Jacquet (France)
3: Metrology and Quality Criteria in Software Measurement	Dr. Asma Sellami—University of Sfax (Tunisia)
4: Quantification and Measurement are not the Same.	Dr. Jean-Marc Desharnais—Ecole de technologie supérieure (Canada) & Bogaziçi University (Turkey)
5: The Design of a Software Measurement Method	Dr. Naji Habra—Facultés Universitaires Notre-Dame de la Paix—FUNDP, Namur (Belgium)
6: Cyclomatic Complexity Number: Analysis of its Design	Dr. Naji Habra—FUNDP (Belgium)
7: Halstead's Metrics: Analysis of their Designs	Dr. Rafa Al-Qutaish—Alain University of Science and Technology, Abu Dhabi Campus, United Arab Emirates
8: Function Points: Analysis of their Design	
9: Use Case Points: Analysis of their design	Joost Ouwerkerk—Expedia (Canada)
10: ISO 9126: Analysis of its Quality Models and Measures	Dr. Rafa Al-Qutaish—Alain University of Science and Technology, Abu Dhabi Campus, United Arab Emirates
11: COSMIC—Design of an Initial Prototype	D. St-Pierre, Dr. Desharnais, Dr. P. Bourque and M. Maya (École de technologie supérieure—University of Québec—Canada)

Chapter	Co-Contributor
12: COSMIC—Scaling up and Industralization	C. Symons, M. O'Neil, P. Fagg, and a number of the COSMIC members of the measurement practices committee
13: Convertibility Across Measurement Methods	Dr Desharnais—Ecole de technologie supérieure (Canada) & Bogaziçi University Turkey
14: Design of Standards Etalons: The Next Frontier in Software Measurement	Dr. Adel Khelifi—University Al Hosn (United Arab Emirates)

Above all, this book is dedicated to all those who provided me with feedback and insights on software measures over the years and who are contributing, each in his or her own way, to the improvement of software measures as a foundation for sound, quantitatively-based decision making.

10

ISO 9126[1]: ANALYSIS OF QUALITY MODELS AND MEASURES

This chapter covers an analysis of ISO 9126 from multiple perspectives:

- The analysis models in ISO 9126 as standard reference models.
- The metrology related part of ISO 9126: the base and derived measures.
- Analysis of the designs of the measurement of Effectiveness in ISO 9126.
- The missing links from Metrology to the Analysis Models.
- Advanced Readings 1: Analysis of the design of the Productivity measure in ISO 9126.
- Advanced Readings 2: Measurement issues associated with the attributes and related base measures in ISO 9126.

10.1. INTRODUCTION TO ISO 9126

In 1991, the ISO published its first international consensus on the terminology for the quality characteristics for software product evaluation: ISO 9126—*Software Product Evaluation—Quality Characteristics and Guidelines for their Use*. From

[1]See also: Alain Abran, Rafa A1 Qutaish, and Juan Cuadrado-Gallego, "Investigation of the Metrology Concepts within ISO 9126 on Software Product Quality Evaluation," 10th WSEAS International Conference on Computers, Athens, Greece, July 13–15, 2006, pp. 864–872.

Software Metrics and Software Metrology, by Alain Abran

2001 to 2004, the ISO published an expanded four-part version, containing both the ISO quality models and inventories of proposed measures for these models.

These ISO 9126 quality models provide by far the most comprehensive coverage in software engineering on software quality:

- Both national and international standards experts have invested a great deal of energy over the past two decades to improve them, and are still working to improve them further.
- These ISO quality models are considerably more detailed than the quality models proposed since the 1980s by individual authors.

The current version of the ISO 9126 series consists of:

- one International Standard (ISO 9126 Part 1) which documents the structure of the ISO quality models for software products, and
- three Technical Reports (ISO TR 9126 Parts 2 to 4):
  - Part **2** proposes a list of derived measures of **external** software quality.
  - Part **3** proposes a list of derived measures of **internal** software quality.
  - Part **4** proposes a list of derived measures for the **quality in use** model.

This is complemented by a set of guides in the ISO 14598 series:

- Developers Guide
- Evaluators Guide
- Evaluation Modules, etc.

The ISO 9126 quality models are also gradually penetrating software organizations in a number of countries. That said, practitioners and researchers alike still encounter major difficulties in the use and exploitation of these quality models, and the ISO itself is putting a great deal of effort into further improvements.

- For instance, within the next two to three years, the ISO expects to issue a major update of this series, which will be relabeled the ISO 25000 series on software product quality.

The analyses presented in this chapter can be useful to both practitioners and researchers:

- For analyzing the foundations of the measures proposed to them.
- For designing the measures themselves, when required, but avoiding the pitfalls inherent in a number of them, including the ISO models.

This chapter is organized as follows:

- Section 10.2 presents the analysis models of ISO 9126.
- Section 10.3 presents the metrology part: base and derived measures.
- Section 10.4 present an analysis of some of the design of the measurement of "Effectiveness" in ISO 9126-4.
- Section 10.5 identifies some of the missing links between the measures and the quality models.
- Section 10.6 identifies an improvement strategy.
- The Advanced Readings 1 section presents an analysis of the design of the Productivity measure in ISO 9126-4.
- The Advanced Readings 2 section presents an illustration of measurement design issues identified in the analysis of the attributes and base measures in ISO 9126.

10.2. ANALYSIS MODELS OF ISO 9126: THE (QUANTITATIVE) MODELS

10.2.1 The Standard Reference Model

The first document in the ISO 9126 series—*Software Product Quality Model*—contains what can be considered as the **ISO Standard Reference Model** for the quality of software products (see also Chapter 4, section 5 and Figures 4.2 and 4.4 for the concept of "standard reference model").

This reference model includes three views of the quality of a software product at the highest level—see Figure 10.1:

- Internal quality of the software
- External quality of the software
- Quality in use of the software

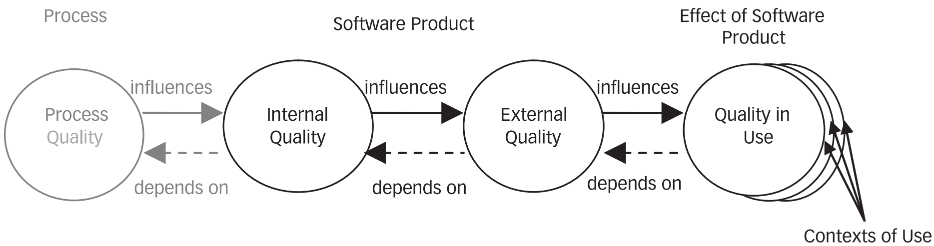

Figure 10.1. Quality in the software lifecycle—ISO 9126-1. This figure was adopted from Figure 2 (p.3) of ISO/IEC 9126-1:2001(E). This figure is not to be considered an official ISO figure nor was it authorized by ISO. Copies of ISO/IEC 9126-1:2001(E) can be purchased from ANSI at http://webstore.ansi.org.

Next, ISO 9126-1 presents two structures of quality models for software product quality:

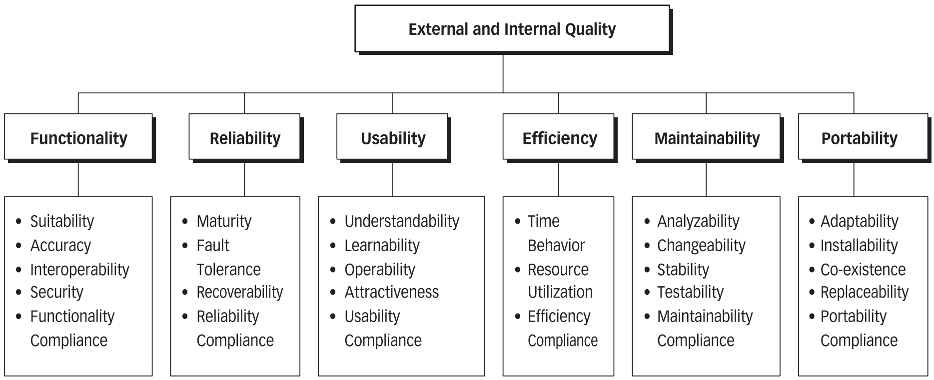

Figure 10.2. Quality model for External and Internal Quality: characteristics and sub-characteristics—ISO 9126-1. This figure was adopted from Figure 4 (p.7) of ISO/IEC 9126-1:2001(E). This figure is not to be considered an official ISO figure nor was it authorized by ISO. Copies of ISO/IEC 9126-1:2001(E) can be purchased from ANSI at http://webstore.ansi.org.

- a 1st structure for both the internal and external quality models—see Figure 10.2, and
- a 2nd structure for the quality in use model.
- The 1st structure (Figure 10.2) includes six characteristics, subdivided into 27 sub characteristics for internal and external quality.
- The 2nd structure includes four "quality in use" characteristics: effectiveness, productivity, safety, and satisfaction.

It must be noticed that ISO 9126 does not provide any reference values for any of its quality characteristics and sub-characteristics.

10.2.2 An Organizational Reference Context Model: Interpretation of ISO 9126 for Evaluation & Decision Making

For the application of the ISO quality models to specific software within an organization, an Organizational Reference Context would typically be set up and used. How to do this is described in the ISO 14598 series, from various perspectives (developers, third party, etc.).

The application of this analysis model (which corresponds to evaluation and decision making on the basis of decision criteria in ISO 9126) is usually performed as a four-step quantification process:

1. Identification of quality-related requirements, that is, the selection of the parts of the ISO quality models that are relevant to a particular context of quality evaluation.
2. Identification of the context of interpretation, that is:
   - the selection of reference values, such values being either generic or specific threshold values, or
   - the determination of targets specified for a particular context.

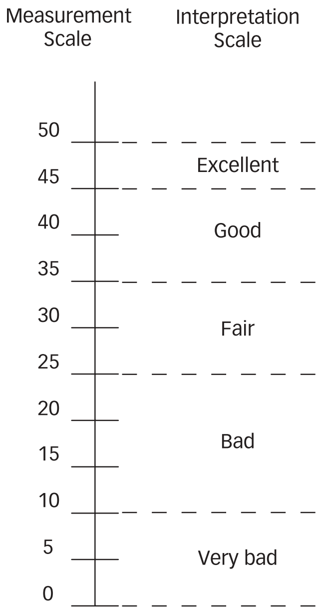

Figure 10.3. Reference values with a measurement scale and an interpretation scale

3. Use of the derived measures from the data preparation phase to fill out the instantiated quality model determined in Step 1.
4. Comparison of the results of Step 3 with either the set of reference values or the targets determined in Step 2 to take a decision based on both the information provided and whatever relevant information is available to the decision maker.

This is illustrated in Figure 10.3, with:

- reference values for the decision criteria (or evaluation criteria) for this **organizational context**
- target values (or an evaluation scale) **for the specific software to be measured**

For the set of relationships over the set of objects of interest for the Information Needs (see ISO 15939 and Chapter 4), the Analysis Model would typically either:

1. quantify a relationship which is well understood, well described over time, and for which there is a large consensus, or
2. attempt to “quantify” a relationship (i.e. a concept) for which it is not yet well known how to capture it within a single measurement dimension and

a single (base or derived) measure (with its corresponding single measurement unit, or set of such units).

While the ISO 9126 quality models are described at a very high level, the relationships across the models, the quality characteristics and sub-characteristics are definitively not well understood and not well described.

Therefore, to use in practice any one of such relationships described textually in ISO 9126 represents an "attempt to quantify" without a prescribed standard or organizational reference context nor empirically verified foundation [Abran *et al.* 2005a, 2005b]. Some of the related issues are described next.

Also, neither of ISO 9126 and ISO 14598 (and the upcoming ISO 25000 series) proposes specific "reference models of analysis" nor an inventory of "organizational reference contexts" with reference values and decision criteria. Each organization has to define its own, thereby limiting possibilities of comparisons industry-specific wide or industry-wide.

10.3. THE METROLOGY-RELATED PART OF ISO 9126: BASE AND DERIVED MEASURES

The implementation of analysis models, including the ones from ISO 9126, has to be based on the data collection of base measures (and derived ones, where relevant).

The measures available for the data collection for the ISO 9126 quality models are proposed and described in the three technical reports in the ISO 9126 series. These reports propose:

- an inventory of +250 derived measures for each quality characteristic or sub characteristic,
- 80 base measures (and corresponding 80 attributes) which are used to build the above +250 derived measures,
- explanations of how to apply and use them, and
- some examples of how to apply them during the software product life cycle.

A key question, then, is: what is the quality, from a measurement perspective, of this extensive list of 250+ proposed measures? Put differently, should practitioners trust and use these proposed measures?

Derived Measures in ISO 9126—2001+

Each of the over 250 derived measures proposed in ISO 9126 is typically defined as a formula composed of base measures.

An example from ISO 9126 Part 4: Quality in use:

- Base Measure 1 (B1): Number of detected failures.
- Base Measure 2 (B1): Number of test cases performed.
- Derived Measure: B1 / B2 with the following measurement units:

$$\frac{B1}{B2} = \frac{\text{Number of detected failures}}{\text{Number of test cases performed}}$$

In this example from ISO 9126, the derived measure (with the corresponding measurement units above) is assigned the following name: "**Failure density against test cases**."

However, while the numerical assignment rules (that is, the formula for the derived measures) for each derived measure are described as mathematical operations (here, a division), neither B1 nor B2 for these operations is described with sufficient clarity to ensure the quality (accuracy, repeatability, and reproducibility) of the measurement results of:

- these base measures, and, of course,
- the derived measures built upon them.

In general, this is a fairly weak foundation from a measurement perspective, and both practitioners and researchers should be very careful in the implementation, use, and interpretation of the numbers that result from the use of these ISO quantitative models.

Notwithstanding the above comments, these ISO documents remain at the forefront of the state of the art on software measurement, and efforts are under way to improve them.

A review from the metrology perspective of the designs of a derived measure in ISO 9126 is presented next and illustrates some of their weaknesses as measurement methods from a metrology perspective.

10.4. ANALYSIS OF DERIVED MEASURES

10.4.1 Analysis of the Derived Measures in ISO 9126-4: Quality in Use

The ISO 9126-4 technical report on the measures proposed for the ISO **Quality in use** model is used to illustrate some of the metrology-related issues that were outstanding in the ISO 9126 series in the late 2000s. Many of the measurement issues raised with respect to ISO Part 4 would also apply to Parts 2 and 3.

In ISO 9126-4, 15 derived measures are proposed for the 4 quality characteristics of the ISO Quality in use model—see Table 10.1.

The objective of the analysis is to enable us to identify the measurement concepts that were not tackled in the ISO 9126 series of documents, that is, their gaps in their measurement designs.

TABLE 10.1. Derived Measures in ISO 9124-4: Quality in Use

Quality Characteristic	Derived Measures
Effectiveness	— task effectiveness — task completion — error frequency
Productivity	— task time — task efficiency — economic productivity — productive proportion — relative user efficiency
Safety	— user health and safety — safety of people affected by use of the system — economic damage — software damage
Satisfaction	— satisfaction scale — satisfaction questionnaire — discretionary usage

- Each of these gaps in the design of the derived measures represents an opportunity to improve the measures in the upcoming ISO update, which is the ISO 25000 series.
- This analysis provides an illustration of the improvements that are needed to many of the software measures proposed to the industry.

10.4.2 Analysis of the Measurement of Effectiveness in ISO 9126

In ISO 9126-**4**, it is claimed that the proposed three measures for the **Effectiveness** characteristic—see Table 10.1—assess whether or not the task carried out by users achieved the specific goals with accuracy and completeness in a specific context of use.

This sub-section identifies a number of issues with:

1. the base measures proposed,
2. the derived measures,
3. the measurement units,
4. the measurement units of the derived quantities, and
5. the value of a quantity for Effectiveness.

10.4.2.1 Identification of the* Base Measure *of Effectiveness. The 3 Effectiveness-derived measures (task effectiveness, task completion, and error frequency) come from a computation of four base measures, which are themselves collected/measured directly, namely:

- task time,
- number of tasks,
- number of errors made by the user, and
- proportional value of each missing or incorrect component.

The first three base measures above refer to terms in common use (i.e. task time, number of tasks, and number of errors made by the user), but this leaves much to interpretation on what constitutes, for example, a task.

Currently, ISO 9124 does not provide a detailed measurement-related definition for any of them:

- In summary, it does not provide assurance that the measurement results are repeatable and reproducible across measurers or across groups measuring the same software, or across organizations either, where a task might be interpreted differently and with different levels of granularity.
- This leeway in the interpretation of these base measures makes a rather weak basis for both internal and external benchmarking.

The third base measure, number of errors made by the user, is defined in Appendix F of ISO TR 9126-4 as an "instance where test participants did not complete the task successfully, or had to attempt portions of the task more than once."

This definition diverges significantly from the one in the IEEE Standard Glossary of Software Engineering Terminology, where the term "error" has been defined as "the difference between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. For example: a difference of 30 meters between a computed result and the correct result.

The fourth base measure, referred to as the "proportional value of each missing or incorrect component" in the task output is based, in turn, on another definition, whereas each "potential missing or incorrect component" is given a weighted value A_i based on the extent to which it detracts from the value of the output to the business or user.

This embedded definition itself contains a number of subjective assessments for which no repeatable procedure is provided:

- the value of the output to the business or user,
- the extent to which it detracts from that value,
- the components of a task, and
- potential missing or incorrect components.

10.4.2.2 The* Derived Measures *of Effectiveness. The proposed three derived measures for the Effectiveness characteristic, which are defined as a prescribed combination of the base measures mentioned above, inherit the weaknesses of the base measures of which they are composed. In summary, there is no assurance that the measurement results of the derived measures are repeatable and reproducible across measurers, across groups measuring the same software, or across organizations either, where a task might be interpreted differently and with different levels of granularity.

***10.4.2.3 The Measurement Units of the* Base Measures.** Of the four base measures, a single one, i.e. task time, has:

- an internationally recognized standard measurement unit: the second, or a multiple of this unit;
- a universally recognized corresponding symbol: "s" for the second as a measure of time.

The next two base measures (tasks and errors) do not refer to any international standard of measurement and must be locally defined. This means that:

- They are not reliably comparable across organizations.
- They are also not reliably comparable within a single organization when measured by different people, unless local measurement protocols (i.e. measurement procedure) have been clearly documented and rigorously implemented.

The fourth base measure (proportional value of each missing or incorrect component) is puzzling:

- it is based on a given weighted value (number), and
- it has no measurement unit.

***10.4.2.4 Measurement Units of the* Derived Measures.** *Task effectiveness:* In ISO 9126-4, this derived measure leads to a derived unit that depends on a given weight:

$$\text{Task effectiveness} = 1 - (\text{a given weight})$$

Therefore, its derived unit of measurement is unclear and undefined.

Task completion: The derived measure is computed by dividing one base measure by the other (task/task) with the same unit of measurement. The measurement results is a percentage.

Error frequency: The definition of the computation of this derived measure provides two distinct alternatives for the elements of this computation. This can lead to two distinct interpretations:

- errors/task, or
- errors/second.

Of course,

- this, in turn, leads to two distinct derived measures as a result of implementing two different measurement functions (formulae) for this same derived measure;
- and leaves open the possibility of misinterpretation and misuse of measurement results when combined with other units. For example: measures in centimeters and measures in inches cannot be directly added or multiplied.

> In software measurement, who cares about this mixing of units?
> Should you care as a software manager?
> Should you care as a software engineer?

10.4.2.5 Value of a Quantity for Effectiveness. The five types of metrology values of a quantity are[2]:

- Numerical quantity value
- Quantity-value scale
- Ordinal quantity-value scale
- Conventional reference scale
- Ordinal value

In the measurement of Effectiveness with ISO 9126-4, for each *base measure* numerical values are obtained on the basis of the defined data collection procedure:

For each *derived measure*, numerical values are obtained by applying their respective measurement function. For instance, the derived measures task effectiveness and task completion are expressed as percentages, and are interpreted as the effectiveness and completion of a specific task respectively.

- For task effectiveness in particular, it would be difficult to figure out both a true value and a conventional true value.
- For task completion and error frequency, the true values would depend on locally defined and rigorously applied measurement procedures, but without reference to universally recognized conventional true values (as they are locally defined).

Finally, in terms of the metrological values:

- Only task time refers to a conventional reference scale, that is, the international standard (etalon) for time, from which the second is derived.

[2]See Table 3.1 in chapter 3 on metrology.

- None of the other base measures in these derived measures of effectiveness refers to a conventional reference scale, or to a locally defined one.

See the Advanced Readings section for an additional example.

10.5. THE MISSING LINKS: FROM METROLOGY TO QUANTITATIVE ANALYSIS

10.5.1 Overview

In ISO 9126, there is an *implicit* claim that the proposed (base or derived) measures capture the intended concept to be quantified. However, there is no *explicit* claim about how much each of the proposed measure quantitatively captures the concept intended to be quantified.

At best, each proposed measure in ISO 9126 is considered as an unspecified contributor to such a quantification for evaluation and decision-making. Put differently, they are:

- not considered as a contributor to concept measurement in terms of a specific scale and a standard etalon,
- but only as a contributor (in some proportion) to its quantification (without an agreed-upon and widely recognized reference scale and a standard etalon).

In other words, in the ISO 9126 analysis models, the practical solution (when a decision criterion cannot be measured directly) is to quantify through an association or mapping (direct or indirect), with, it is hoped, better understood concepts that have been quantified.

An example is presented next with some of the ISO 9126 base measures combined into derived measures which are associated with other concepts (i.e. the quality characteristics and sub-characteristics within the ISO model of software product quality).

This section illustrates the missing links between the quantitative models proposed in ISO 9126 and the detailed measures proposed as the contributors to this quantification. The example selected is that of the **Maturity** sub-characteristic and of the 7 derived measures proposed for this sub-characteristic.

For the **Maturity** sub-characteristic as an example:

- The relationships between the attribute being measured by the derived measure "failure density against test cases" and the "maturity" sub-characteristic and the "reliability" characteristic are not identified, neither described.
- No model of such relationships is proposed either.

10.5.2 Analysis of the Measurement of "Maturity"

The model proposed by ISO to quantify the external quality of a software product contains—as previously seen in Figure 10.2—six quality characteristics and 27 sub-characteristics:

- One of the 6 quality characteristics is "**Reliability**" which is itself composed of 4 sub-characteristics.
- One of these 4 sub-characteristics is "**Maturity**" which is chosen here as an example.

To quantify this single **Maturity** sub-characteristic, ISO 9126-3 proposes a list of seven distinct derived measures:

1. Failure density against test cases
2. Estimated latent fault density
3. Fault Density
4. Fault removal
5. Mean time between failures (MTBF)
6. Test coverage
7. Test maturity

Related ISO 9126 Definitions

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

Maturity:
The capability of the software product to avoid failure as a result of faults in the software.

10.5.2.1 Metrology Perspective. Each of these seven derived measures is presented only at a fairly abstract level as formulae composed from a set of base measures, themselves lacking detailed descriptions, including the attributes they are attempting to measure.

This leaves each of them highly susceptible to individual interpretation: neither the base measures for these operations, nor the corresponding attributes, have been described with sufficient clarity to ensure the quality of the measurement results.

They are not documented at a detailed enough level to provide sufficient guidance to ensure the accuracy, repeatability, and repetitiveness of measurement results, in the event that the same software is measured by different measurers, which in turn leads to potentially significantly different values.

TABLE 10.2. A Derived Measure Proposed for Quantifying Maturity in ISO 9126-2: Failure Density Against Test Cases

Step (ISO 15939)	Description
Data Collection:	• Base Measure 1 (B1): Number of detected failures. • Base Measure 2 (B1): Number of test cases performed.
Data Preparation	• Algorithm of Derived Measure: B1 / B2 • Name of Derived Measure: Failure density against test cases.
Analysis	The derived measure, Failure density against test cases, is associated * with (mapped to): • The quality sub-characteristic: Maturity • Within the quality characteristic: Reliability • Within the quality sub-model: External Quality—ISO 9126-2 * Associated: there is not an explicit description in ISO 9126 of how much of the proposed measure quantitatively captures the concept intended to be quantified.

Each one of the seven proposed derived measures are described individually as illustrated in the side box and Table 10.2 with the "Failure density against test cases" as an example.

Example: "Failure Density Against Test Cases"

Purpose of this derived measure in ISO 9126: how many failures were detected during defined trial period?

Method of application for this derived measure: count the number of detected failures and performed test cases.

However, none of the embedded base measures are defined precisely in ISO 9126, including failure and test cases.

10.5.2.2 Perspective of the Analysis Model in ISO 9126. There is no ISO defined context or values for such context. Each group within each organization has to build its own set of values for analysis within a specific context.

- What is the specific contribution of any one of the above seven derived measures to the Maturity sub-characteristic?
- Are there some overlaps across the relationships of any of these seven derived measures, and if so, to what extent?
- If not all seven derived measures are mandatory-necessary, which one or which ones are the most representative of the Maturity sub-characteristic, and if so, to which extent?

We now ask how well the derived measure, "Failure density against test cases," fully describes (i.e. quantifies) the above hierarchy of concepts of the ISO 9126 model of software product quality:

How much of the Maturity sub-characteristic is captured by it?

- In practice, the derived measure, Failure density against test cases, is only a contributor, that is, an indicator—see definition of "indicator" in side box in Section 4.5.1 of Chapter 4—within that piece of the hierarchy of quality-related concepts.
- How can we quantify this contribution? The answer is not documented in ISO 9126-4, or even identified as an issue.

10.5.2.3 Missing Links. None of the expected links between this (weak) metrology basis for the measurement of the base and derived attributes and the quantification of the quality sub-characteristic (e.g. Maturity) and characteristic (e.g. Reliability) is described in ISO 9126.

Hopefully, such links will be described but it will take years of research and development to gain insights into this problem and to develop analysis models based on solid empirical evidence.

10.6. OUTSTANDING MEASUREMENT DESIGN ISSUES: IMPROVEMENT STRATEGY

As has been illustrated throughout this chapter, the base measures and their current lack of adherence to metrology principles and characteristics constitute one of the major impediments to the overall applicability and deployment of ISO 9126.

It is also recognized that, to properly analyze, verify, and correct the design of all 80 base measures will require considerable time and work on the part of a single, isolated research team.

To tackle this challenge, a larger, multi-group research team has to be set up, its members to work concurrently on this issue, including research groups at universities across the world, all of whom in coordination with the work in progress of ISO/IEC JTC1/SC7 Working Group 6. Organizing the work in this way should facilitate the transition of the research results to the ISO normative level and, consequently, enable quicker integration of adequately structured base measures into the ISO 25000 series of standards for the measurement of the quality of software products.

In particular, improvement work should focus on five of the 80 attributes in ISO 9126 which appear in more than 10 derived measures; that is, function (38 occurrences), duration (26), task (18), case (16), and failure (11)—See the Adanced Readings 2 section.

- Work on the detailed design of the base measures and on the definitions of the attributes should leverage relevant measurement definitions from other international standards wherever possible.

- Even definitions from existing standards still need further refinement to facilitate their use in operational procedures from a measurement viewpoint.
- Finally, the ISO 9126 standard also includes a number of qualifiers of the base measures which will require further clarification from a measurement viewpoint.

In conclusion, much work remains to be done:

- To define the base measures in detail, even those identified as requiring priority attention.
- To identify the links across the quality models and their corresponding base and derived measures proposed in ISO 9126.

ADVANCED READINGS 1: ANALYSIS OF THE DESIGN OF THE PRODUCTIVITY MEASURE IN ISO 9126

Audience

This Advanced Reading section is of most interest to those interested in additional examples of difficulties currently found in many of the measurements proposals to industry.

This includes measures proposed in some ISO documents that are not yet approved as International Standard, but only on an interim basis as ISO Technical Reports such as ISO 9126 Technical Reports 2 to 4.

In ISO 9126-4, a claim is made that the five derived measures of the **Productivity** characteristic—see Table 10.1—assess the resources that users consume in relation to the effectiveness achieved in a specific context of use. The time required to complete a task is considered to be the main resource to take into account in the measurement of the Productivity characteristic of quality in use.

Of the five proposed measures of **productivity** in ISO 9126-4, one is a base measure: task time. The other four are derived measures:

- task efficiency,
- economic productivity,
- productive portion, and
- relative user efficiency.

It is to be noted that task efficiency refers explicitly to another derived measure, task effectiveness, which was analyzed in a previous section.

It is also to be noted that these four *derived measures* are themselves based on five *base measures*:

- task time,
- cost of the task,
- help time,
- error time, and
- search time.

Measurement Units of the derived measures

The measurement unit for *task efficiency* is not completely clear, since it depends on an ill-defined "given weight:"

$$\begin{aligned}\text{“task efficiency” unit} &= \frac{\text{“task effectiveness” unit}}{\text{second}}.\\ &= \frac{1 - \text{“a given weight” unit}}{\text{second}} = \frac{?}{\text{second}}.\end{aligned} \tag{1}$$

Similarly, the measurement unit of *economic productivity* depends on the measurement unit of task effectiveness, a derived measure, which is unknown:

$$\begin{aligned}\text{“economic productivity” unit} &= \frac{\text{“task effectiveness” unit}}{\text{currency unit}}\\ &= \frac{1 - \text{“a given weight” unit}}{\text{currency unit}}\\ &= \frac{?}{\text{currency unit}}.\end{aligned} \tag{2}$$

The *productive proportion* is expressed as a percentage and, as such, has no measurement unit (it has the same measurement unit in both the numerator and the denominator):

$$\text{“productive proportion” unit} = \frac{\text{second}}{\text{second}}. \tag{3}$$

Finally, *relative user efficiency* has no measurement unit either, since the measurement units in both the numerator and the denominator are the same here as well (the task efficiency measurement unit), and therefore the result of this derived measure is also expressed as a percentage:

$$
\begin{aligned}
\text{“relative user efficiency” unit} &= \frac{\text{“task efficiency” unit}}{\text{“task efficiency” unit}} \\
&= \frac{\dfrac{\text{“task effectiveness” unit}}{\text{second}}}{\dfrac{\text{“task effectiveness” unit}}{\text{second}}} \\
&= \frac{\dfrac{1-\text{“a given weight” unit}}{\text{second}}}{\dfrac{1-\text{“a given weight” unit}}{\text{second}}} \\
&= \frac{\dfrac{?}{\text{second}}}{\dfrac{?}{\text{second}}}. \qquad (4)
\end{aligned}
$$

ADVANCED READINGS 2: ATTRIBUTES AND RELATED BASE MEASURES WITHIN ISO 9126

Audience

This Advanced Reading section is of most interest to researchers and industry practitioners interested in improving measurement standards such as those of the ISO 9126 and upcoming ISO 25000 series, and in figuring out priorities in the selection of the base measures to be improved.

Introduction

An inventory of the base measures in ISO 9126 has identified 80 distinct base measures [Desharnais *et al.* 2009]: Have the attributes to be measured in ISO 9126 by these 80 base measures been described with sufficient clarity to ensure the quality of the measurement results?

Improving the design of these 80 base measures is essential for the use of the ISO models in industry. To do so is a daunting task, considering the number of steps and iterations typically necessary to design software measures adequately, as illustrated in Part 1 of this book (Chapters 2 to 5):

- This design task is even more challenging when, in addition to the view of the person designing a measure, a consensus must be developed progressively at an international level, such as within an ISO committee composed of domain experts from a number of countries.
- Similarly, to determine which of these base measures must be improved in the timeliest fashion is a challenge.

A framework to define base measures

In terms of improving the measurement foundation of ISO 9126, the main interest is in the first phase, the design of the measurement method, which includes the following activities:

1. Defining the measurement principle where this activity gives the precise description of what is going to be measured.
2. Defining a measurement method on the basis of that principle, an activity which gives a general description of how to measure.
3. Determining an operational measurement procedure; that is, an implementation of the method in a particular context, an activity which gives a detailed description of how to measure.

It can be observed that, for most of its *base measures*, the ISO 9126 standard:

- does not provide a precise definition of the attribute being measured, i.e. the "what,"
- does not provide a generic description of how to measure that attribute, and
- does not provide operational measurement procedures.

For the upcoming version of ISO 9126 in 2010+ (i.e. ISO 25021), the "what" and the "how" should be spelled out by going through the above three steps to improve the design of the base measures.

- In the late 2000s, only a very small number of software base measures in the whole set of software engineering standards, such as those for the measurement of software functional size, have already gone through all these steps.
- Most in ISO 9126 do not even have a normalized definition of their attributes, and therefore no precise description of what must be measured.

Defining the full set of 80 attributes and necessary base measures is not considered a task which will be feasible to perform within the next two to three years, even by a large organization such as the ISO, which has access to a large international network of domain experts.

- Which of these 80 base measures and corresponding attributes should be addressed first, and which ones will have the most impact initially?
- Which of these 80 base measures are used most in the ISO 9126 series?

Table 10.3 presents a summary of the distribution of occurrences of the 80 base measures (and corresponding single attribute) within the full set of 250 derived measures in the three Parts of ISO 9126:

TABLE 10.3. Occurrences of Base Measures within the Derived Measures of ISO 9126

Occurrences of the Base Measure (Single Attribute) in Derived Measures	Number of Base Measures (Attributes)	Percentage
More than 10	5	6%
From 3 to 10	15	19%
2	13	16%
1	47	59%
TOTAL	**80**	**100%**

- 5 of the 80 base measures (and corresponding single attribute) appear in more than 10 derived measures,
- 15 base measures occur from 3 to 10 times,
- 13 base measures occur twice, and
- 47 base measures have a single occurrence (or 59% of the 80 distinct base measures).

Defining the attributes with more than 10 occurrences
The five base measures (and corresponding distinct attribute) with more than 10 occurrences:

- function (38),
- duration (26),
- task (18),
- case (16), and
- failure (11).

These five base measures appear between 11 and 38 times.

1. The attribute **function** is consistently used with "number of …" in ISO 9126—Parts 2 to 4.

   However, nowhere is this attribute defined precisely. Therefore, its interpretation in practice can vary considerably across individuals, technology, functional domains, etc.

   Nevertheless, the industry has developed various consensuses over the years on the measurement of the functional size of software. This has led to the adoption of the following international standards for functional size measurement:

   - COSMIC-FFP v2.1 [ISO 19761],
   - IFPUG FPA v4.1 Unadjusted [ISO 20926],
   - Mk-II FPA v1.3.1 [ISO20968],

- NESMA FPA v2.1 [ISO 24570]
- FISMA FPA v1.1 [ISO 29881].

Since these standards have already been recognized by the ISO and are extensively used in practice for productivity and benchmarking studies, they could also be used as normalization factors in quality measurement, such as in the measurement of defect density (that is, the number of defects in software normalized by the functional size of that software, allowing a meaningful comparison of defect density in two distinct pieces of software of different sizes).

2. The other attribute, **case** (with 16 occurrences) is not defined in the ISO 9126 standard. But a "case" is defined as follows in ISO 24765: *"a single-entry, single-exit multiple-way branch that defines a control expression, specifies the processing to be performed for each value of the control expression, and returns control in all instances to the statement immediately following the overall construct."*
3. The attribute **duration** is a length of time in seconds, minutes, hours, etc. The "second" as a unit of measurement is already well defined and is a part of the set of international standards for units of measurement.
4. The attribute **task** has multiple definitions within the ISO standards:
   - a sequence of instructions treated as a basic unit of work by the supervisory program of an operating system, in ISO 24765;
   - in software design, a software component which can operate in parallel with other software components, in ISO 24765;
   - the activities required to achieve a goal, in 4.3 of ISO TR 9126-4;
   - a concurrent object with its own thread of control, in ISO 24765;
   - a term for work, the meaning of which and placement within a structured plan for project work varies by the application area, industry, and brand of project management software, in the PMBOK;
   - required, recommended, or permissible action, intended to contribute to the achievement of one or more outcomes of a process, in section 4.5 of ISO 12207 and in section 4.34 of ISO 15288.

   Therefore, for the task attribute, it is necessary to revise each usage of task for each attribute in each quality characteristic and sub-characteristic.
5. The attribute **failure** is quite challenging, since it has multiple definitions:
   - termination of the ability of a product to perform a required function or its inability to perform within previously specified limits—see 4.2 in ISO 25000;
   - the inability of a system or component to perform its required functions within specified performance requirements—ISO 24765;

- an event in which a system or system component does not perform a required function within specified limits—IEEE 982.1;
- the termination of the ability of a functional unit to perform its required function—IEEE 982.1.

NOTE: A failure may be produced when a fault is encountered.

The first definition of the failure attribute could be suggested, but should be revised in the context of each attribute in each quality characteristic and sub-characteristic.

It should be noted that, even though a number of definitions may exist in the standards (or in the literature) for the attributes mentioned above, this does not necessarily mean that they should be used without further scrutiny:

- the definitions might not have been tested operationally,
- a definition might not be useful in a particular measurement context, or
- there may be multiple definitions of an attribute.

Attribute qualifiers

In addition to the 80 different base measures and over 250 derived measures, ISO 9126—Parts 2, 3, and 4 include a number of qualifiers which characterize some aspects of the base measures (and corresponding distinct attribute). For example, the base measure, number of failures, may refer at times to the number of **resolved** failures or at others to the number of failures **actually detected**. The terms "resolved" and "actually detected" are referred to here as qualifiers of the term "failures" for this base measure; that is, they qualify a subset of the same attribute.

Sometimes the qualification of the base measure uses a broader qualifier. For example, the number of *critical and serious* failure *occurrences avoided*.

Most of the time, the qualifiers in the ISO 9126 quality model are added to measures using a sentence, not just a word.

A solution would be to suggest, whenever possible, a reference in the set of ISO standards.

Example: Type of Maintenance

Type of maintenance could be aligned, along with its corresponding concepts, with the ISO standard on software maintenance—ISO 14764.

Another possibility, when there is no reference to a standard for specific qualifiers, would be to modify them when relevant. To define the important attributes for the ISO 9126 quality model would represent an important improvement.

After completing the priorities in this first iteration of improvements, further effort will be necessary to define the qualifiers.

EXERCISES

1. Describe and explain the link between the Analysis Models and the derived measures in the ISO 9126 models of software quality.
2. There are 80 base measures in ISO 9126. Name a few that are specific to software.
3. How many of these base measures are related to international standards of measurement?
4. Name a base measure specific to software in ISO 9126 that is supported by a well-documented measurement method?
5. What is the measurement unit of the "task efficiency" derived measure?
6. What is the measurement unit of the "productive production" derived measure?
7. Describe the relation and contribution of the "task efficiency" derived measure to the "productivity" characteristic in ISO 9126-4?

TERM ASSIGNMENTS

1. Use the four derived measures proposed by ISO 9126-4 for the "productivity" characteristic. Compare the measurement results obtained with each of these distinct measures and discuss differences (and similarities) in measurement results. What is the contribution of each to the "productivity" characteristic?
2. Design an analysis model using the above 4 derived measures for the "productivity" characteristic that could be used to evaluate "productivity" and take a decision based on the numerical results.
3. Select one of the base measures from ISO 9126-2 and improve its design.
4. Select one of the base measures from ISO 9126-3 and improve its design.
5. Identify and document a process to progressively develop a much larger consensus on the proposed improved design of one the base measures from ISO 9126.
6. Take any one of "metrics" proposed in Parts 2 to 4 of ISO 9126 and propose a design of a *base* measure.
7. Take the *base* measure designed by one of your classmates and analyze the quality of this design. Also, suggest some improvements to your classmate's design.
8. Take any of the base measures from ISO 9126 designed by one of your classmates (see assignments 6 and 7 above) and build a measurement

procedure to ensure that such a base measure produces measurement results that are accurate, reproducible, and capable of replication.

9. Take any one of the "metrics" proposed in Parts 2 to 4 of ISO 9126 and propose a design for a *derived* measure.

10. Take the *derived* measure designed by one of your classmates and analyze the quality of the design. Also, suggest some improvements to your classmate's design.

11. Take any of the quality sub characteristics in ISO 9126 and discuss the methodology you would use to describe the linkages between the sub characteristic selected and one of the metrics actually described in ISO 9126.

12. Take any of the derived measures from ISO 9126 and identify sources of uncertainty in their measurement process.

13. Propose an ISO 9126 quality model for Web software in your organization and provide:

- the rationale for the quality characteristics and sub characteristics selected;
- the rationale for the determination of the evaluation scales for each;
- the rationale for the determination of the target values for each.

14. Assess the maintainability of a piece of software by using a software source code assessment tool (for example, Checkstyle, Logiscope, etc.). Identify which of the ISO9126 maintainability measures can be assessed.