

Academic year 2025 / 2026

**Chapter 08:
Quality evaluation models of software
products**

Presented By: M.Halassa

Introduction

2

➤ Evaluating quality consists of defining rules, interpreting quality indicators according to these rules, giving them meaning and using them to qualify qualitative concepts.

➤ To achieve this, companies use quality models. These models define concepts, domains and qualitative rules that they evaluate via measurements and metrics performed on the system.

What is a standard?

3

- **Standards are documented agreements** containing technical specifications or other specific criteria intended to: be used consistently as rules, guidelines or feature definitions to ensure that materials, products, processes and services are fit for: their use.

Why international standardization

4

- The existence of non-harmonized standards for similar technologies, in different countries or regions, can contribute to: so-called "technical barriers to trade ". Export-oriented industries have long felt the need to agree on global standards to help streamline the **international trade** process. It is precisely this objective that led to: the creation of ISO (**International Organization for Standardization**)

Quality evaluation models of software

5

- We present the main quality models currently in existence.
- These are hierarchical models that identify quality principles starting from global requirements and the most general principles to go down to the metrics that allow them to be measured.

McCall

Metric Criteria Factors - FCM

6

➤ McCall created in 1977 a model to measure the quality level of a software. Its model is composed of three layers: **metrics, criteria and factors**.

➤ **Quality factors** represent the external vision of quality, that of the user. McCall identifies **fifty** factors but only retains **eleven** as the most significant, distributed in three categories:

1. factors related to the operation of the product.
2. factors related to the revision of the product.
3. factors related to product transition

McCall model

Metric Criteria Factors - FCM

7

- **Quality Criteria:** The lower or second-level quality attributes that can be accessed either subjectively or objectively are called Quality Criteria. These attributes are internal. Each quality factor has many second-level quality attributes or quality criteria.
- This model is extremely comprehensive but also very difficult to apply due to the 300 metrics
- This model is also the basis for many other quality models such as **ISO** standards.

Software quality evaluation

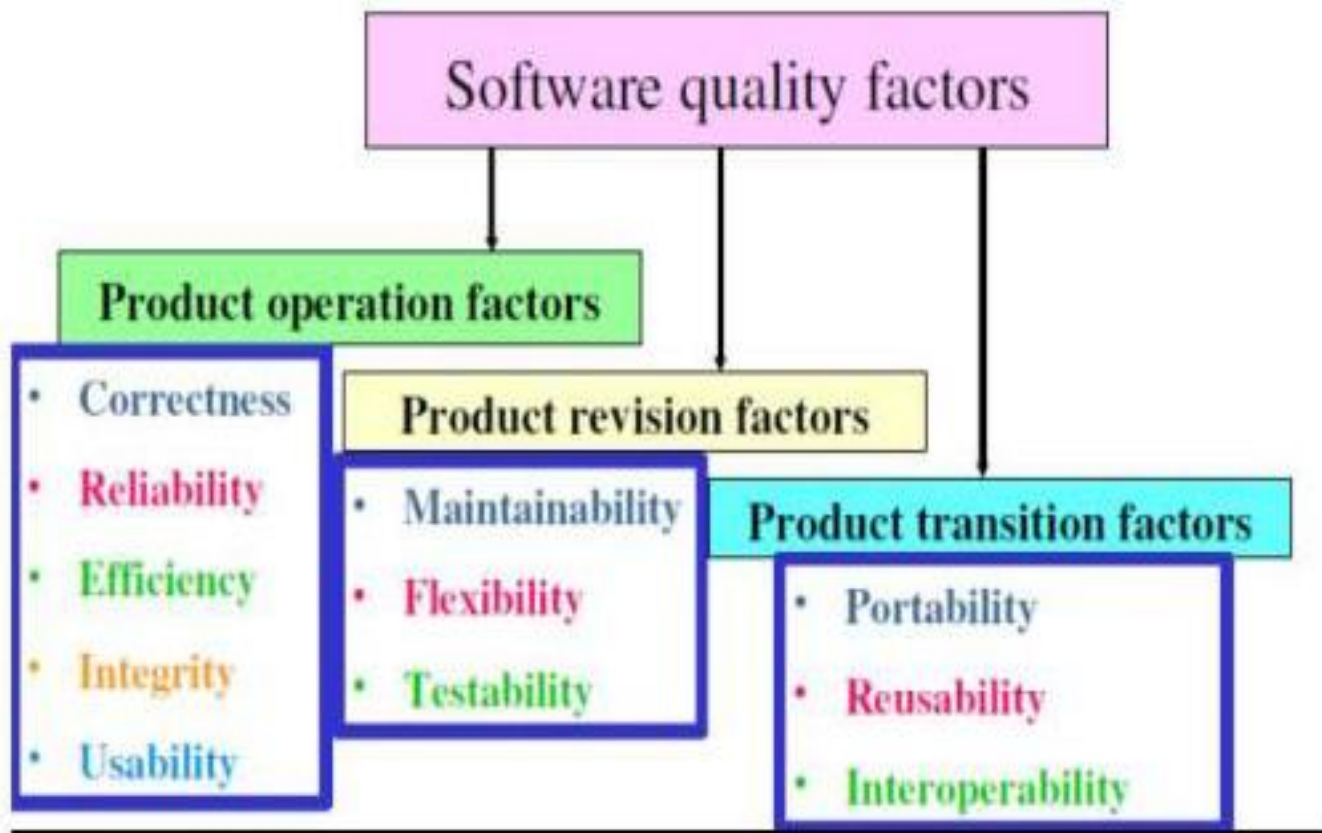
8

- ✓ **A factor:** is a feature of the software, process or service that contributes to its quality as experienced by the user.
- ✓ **A criterion:** is an attribute of the software through which a factor can be obtained. This is also a feature of the software that the developer can act on. (e.g. its simplicity)
- ✓ **A metric:** is the measure of a property of a criterion. (for example, the size of a module for the "Simplicity" criterion).

McCall model

Metric Criteria Factors - FCM

9



McCall model

Metric Criteria Factors - FCM

10

☐ Correctness

- accuracy, completeness of required output
- datedness, availability of the information

☐ Reliability

- maximum failure rate

☐ Efficiency

- resources needed to perform software function

☐ Integrity

- software system security, access rights

☐ Usability

- ability to learn, perform required task

McCall model

Metric Criteria Factors - FCM

11

☐ Maintainability

- effort to identify and fix software failures
(modularity, documentation, etc)

☐ Flexibility

- degree of adaptability (to new customers, tasks, etc)

☐ Testability

- support for testing (e.g. log files, automatic diagnostics, etc)

McCall model

Metric Criteria Factors - FCM

12

☐ Portability

- adaptation to other environments (hardware, software)

☐ Reusability

- use of software components for other projects

☐ Interoperability

- ability to interface with other
components/systems

ISO 9001

13

- ISO 9000 set of recommendations and standards for quality assurance in customer-supplier relations (not specifically software). In particular it includes:
- ISO 9000-1 recommendations for the use of these standards
- ISO 9001 the standard to be used for the supply of software
- ISO 9003 guide for the use of ISO 9001 standards for the supply of software.

ISO 9001 Philosophy

14

- **ISO 9001 philosophy:** Any operation influencing quality must be under control
- This control must be visible
- Certification makes it possible to obtain wide recognition .

you have to pay.

Valid for three (3) years.

ISO 9001

15

- ❖ **-Quality elements:** Chapter 4 of the standard defines the 20 quality elements to be respected:
- ❖ Management Responsibilities
- ❖ Definition of a quality system
- ❖ Analysis of the contract between customer and supplier
- ❖ Design examination
- ❖ Data control
- ❖ Purchasing and supply specification
- ❖ Customer Supplied Products Control
- ❖ Identification and traceability
- ❖ Process Control
- ❖ Inspection and testing control, test tool measurements

ISO 9001

16

- **Quality elements – continued**
 - ❖ Status of tests and inspections
 - ❖ Need to implement tests and inspections
 - ❖ Control of Nonconforming Product
 - ❖ Corrective and Preventive Actions
 - ❖ Packaging, storage, delivery,
 - ❖ Quality Record Control
 - ❖ Quality Audit
 - ❖ Organize on-the-job training
 - ❖ After-sales service
 - ❖ Statistical techniques

The ISO/IEC 9126 standard

17

- **ISO/IEC 9126.** (ISO: International Standardization Organization): a set of indicators for software quality, and thus "facilitates" the software evaluation process and the specification of functional or non-functional requirements. IEC (International Electrotechnical Commission) This standard has been in application since 1992.
- This standard evolves to be enriched and integrated into the SquaRE standard (Software product Quality Requirement and Evaluation) replaced by SquaRE in 2011,
- ISO/IEC 9126 offers **6 quality characteristics** of the software product.

The ISO/IEC 9126 standard

18



The ISO/IEC 9126 standard

19

1. *Functionality*

Set of attributes relating to the existence of a set of functions and their properties. Functions are those that satisfy **expressed or implied needs**.

The ISO/IEC 9126 standard

20

1. *Functionality*

It has the following characteristics:

- 1. Ability:** Presence and suitability of a series of functions for given tasks
- 2. Accuracy:** provision of fair or agreed results or effects
- 3. Interoperability:** ability to interact with given systems
- 4. Security:** ability to prevent unauthorized access (accidental or deliberate) to programs and data

The ISO/IEC 9126 standard

21

- 2. Reliability** (*Reliability*) A set of attributes relating to **the software's ability** to maintain its level of service under specific conditions and for a specified period of time:
- 1. Maturity:** frequency of failures due to software faults
 - 2. Fault tolerance:** ability to maintain a given level of service in the event of a software fault
 - 3. Recoverability:** ability to restore its service level and restore directly affected information in case of failure;

The ISO/IEC 9126 standard

22

3. *Ease of use* (*Usability*)

This criterion has these subcharacteristics:

1. **Ease of understanding:** effort that the user must make to recognize the logic and its implementation
2. **Ease of learning:** effort that the user must make to learn its application, he must understand what can be done with the software, and know how to do it
3. **Ease of operation:** effort that the user must make to operate and control the operation of his application.

The ISO/IEC 9126 standard

23

4. *Efficiency*: (*Efficiency*)

A set of attributes relating to the relationship between **the service level of software and the amount of resources used**, under determined conditions.

The ISO/IEC 9126 standard

24

4. *Efficiency*: (*Efficiency*)

This criterion has the following subcharacteristics:

- **Behaviour with respect to time:** response and processing times.

- **Behaviour towards resources:** amount of resources used; duration of their use when performing its function

.Software must meet execution time constraints by developing simpler products while ensuring the complexity of algorithms and using more efficient machines.

The ISO/IEC 9126 standard

25

5. Maintainability

This criterion has the following subcharacteristics:

- 1. Ease of analysis:** effort required to diagnose deficiencies and causes of failure or to identify parts to be modified
- 2. Ease of modification:** effort required to modify, remedy defects or change the environment.
- 3. Stability:** risk of unexpected effects of changes
- 4. Ease of testing:** effort required to validate the software

The ISO/IEC 9126 standard

26

- 5. Maintainability** there are three types of maintenance
- **Perfective (scalable)** maintenance that consists of maintaining previous features while adding new features that profoundly change the architecture
 - **Adaptive maintenance** that corresponds to the addition of small features that do not change the architecture
 - **Corrective maintenance** that allows the correction of errors updated by the customer and not during verification and validation tests.

The ISO/IEC 9126 standard

27

6. *Portability*

This criterion has the following subcharacteristics:

- **Ease of adaptation:** possibility of adapting to different given environments.
- **Ease of installation:** effort required to install the software in a given environment
- **Compliance with portability rules:** compliance with standards and conventions relating to portability
- **Interchangeability:** possibility and effort of using the software instead of another given software in the same environment

The ISO/IEC 9126 standard

28



- Process quality: qualité des processus du cycle de vie du logiciel
- Internal quality: qualité des produits intermédiaires, incluant les modèles statiques et dynamiques, la documentation et le code source
- External quality: qualité du système final tel que vérifiable à son comportement externe
- Quality in use: effet du système lors de son utilisation – satisfaction des attentes des utilisateurs

Performance level

Quality in use



Factors (to be specified):

view of the software by the user (external)

Criteria (to be constructed):

the view of the software by the developer (internal)

Metric (to be checked):

Allows you to define a scale to **measure** the quality of the software

The SQuaRE standard (Software product QUALity Requirement and Evaluation)

30

- The SQuaRE standard defined since 2005 from ISO 9126 and the evaluation part of the ISO 14598 standard [ISO/IEC, 2005]. It aims to integrate these two standards to form one. It also wants to meet the different needs of quality according to the actors (developers, testers, users, customers).
 - In addition, it unifies the various normative documents around quality.
- SQuaRE** defines four steps to enable better quality assessment: defining requirements, establishing a quality model, defining quality metrics, and evaluating quality.

The SQuaRE standard (Software product QUALity Requirement and Evaluation)



Difference compared to ISO 9126.

32

- The SQuaRE standard takes the model defined in ISO 9126 and makes the following changes:
 - the scope of the quality model has been extended to include quality in use as a model in its own right (however, there remains the usability factor in the model);
 - the standard defines two quality models: the model referring to quality in production and the one measuring quality in use;

Difference compared to ISO 9126.

33

- security has become a feature instead of a sub-feature of functionality.
- portability has been split into two: on the one hand portability which refers only to ease of installation and interchangeability, and on the other hand compatibility which includes interoperability
- the following subcharacteristics have been added: robustness, utility, technical accessibility, modularity, reusability and portability .
- several characteristics and subcharacteristics have been specified and renamed.