Software Quality Model

DR JULIANA JAAFAR

Learning Objective

To introduce the concept of :-

- Software Quality Model
- The importance of comprehensive requirements
- Criteria of quality requirements
- Software quality model contributes to quality requirements

Product Quality Vs Process Quality

- **Product Quality**
  - The overall quality of the product .i.e. how well it conforms to the product requirements, specifications, and customer expectations and g customer satisfaction by improving products (goods) and making them free from any deficiencies or defects.
- **Process Quality**
  - Defined as all the steps/ activities /process used in the manufacturing the final product.
  - Its focus are on all activities and steps used to achieve a maximum acceptance and getting a working product into the customer's hands.

Software Quality Model

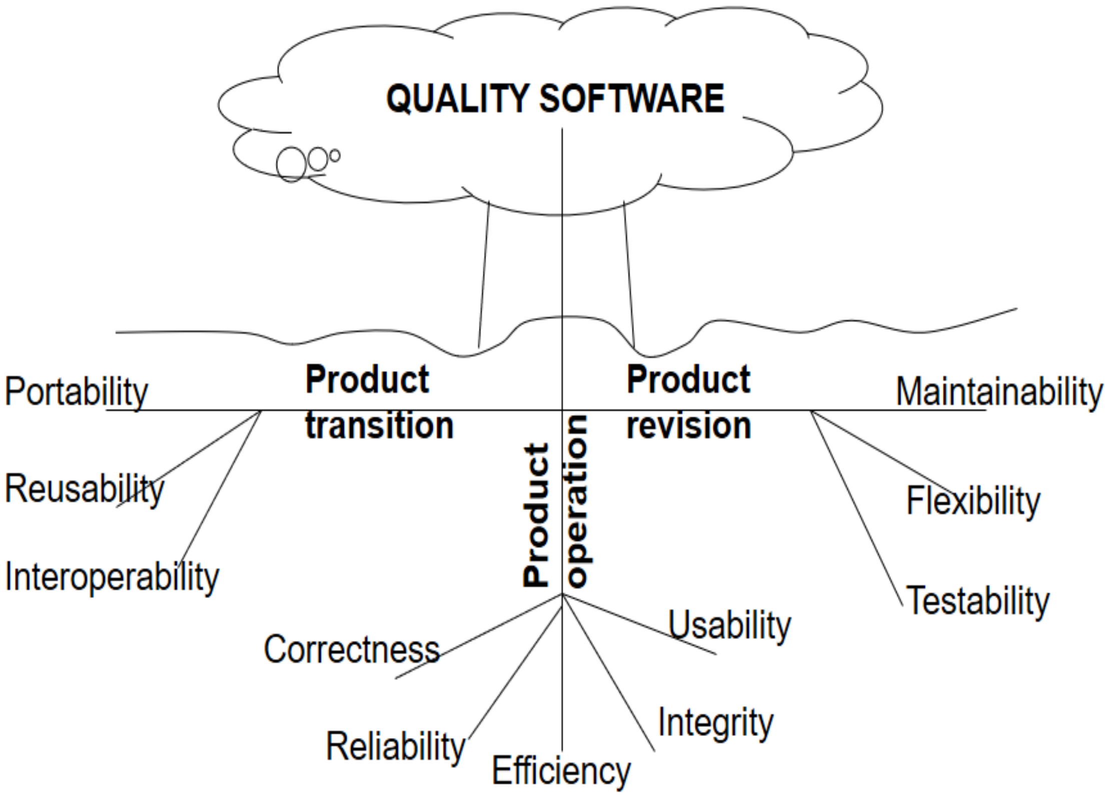
Software quality Model is a standardized way of **understanding the quality concept of a software,** to **measure software systems' quality**, and to **guide software development activities in order to improve software quality**.

- Example of Software Quality Models.
  - Mc Call's Model
  - Boehms Model
  - FURPS Model
  - ISO 9126 Model
  - CMMI Model (Process Improvement)

Other Software Quality Models: Software Quality Models - A comparative Study.pdf

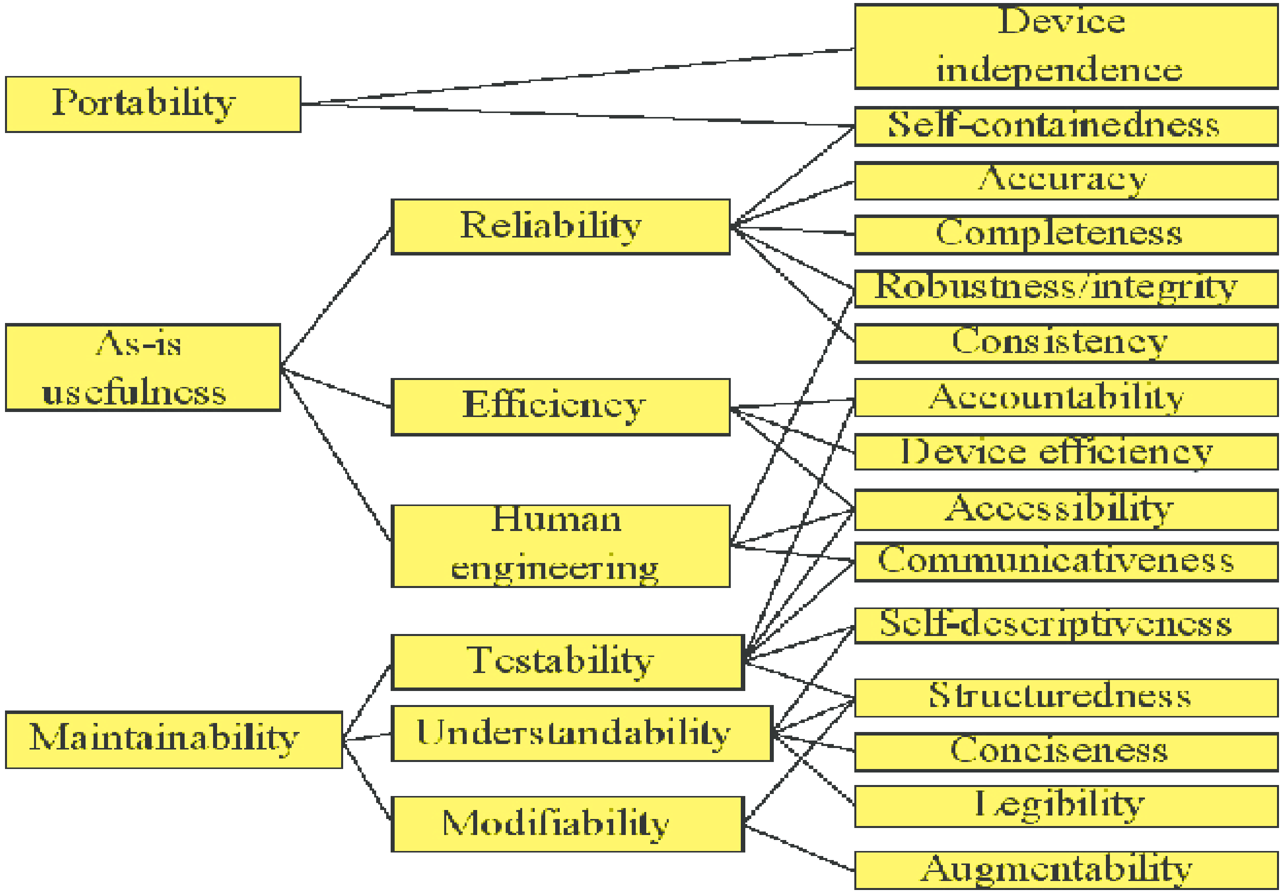
Software Quality Model – **Mc Call's Model**

- The 1st quality model developed by Jim McCall in1977
  - **Product Operation** - The software can run successfully in the market according to the specifications without any defects.
  - **Product Revision** – Quality factors that changes or the ability to change the software product in the future
  - **Product Transition** – Factors that deal with the adaptation of the product to new environments.



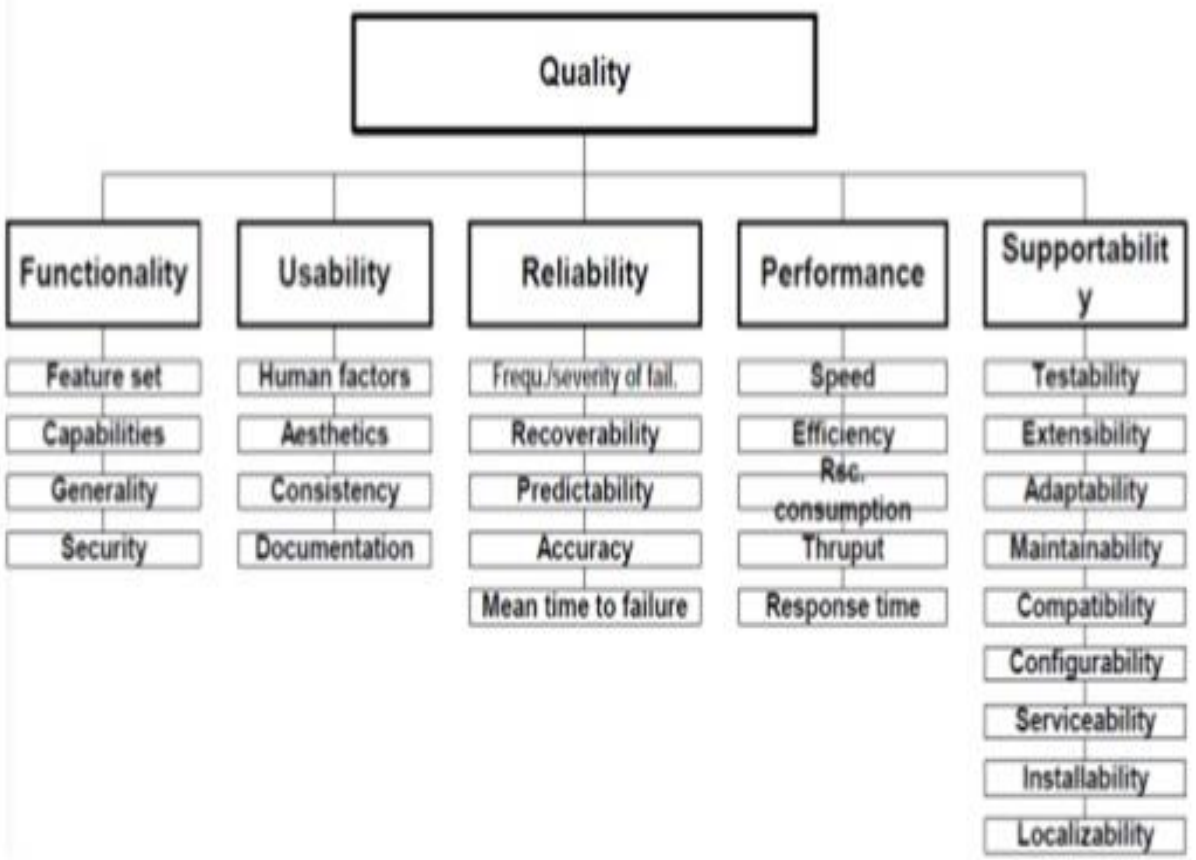
Software Quality Model – **Boehm's Model**

- Claimed to be the improved version of Mc Call Model – Developed by Barry W. Boehm (1978)
  - Included the hardware performance
- The model reflects on
  - What the user want?
  - Uses of resources
  - Is software easy to learn and use?
  - Is well designed, well coded, easily tested and maintained?



Software Quality Model – **FURPS Model**

- Introduced by Robert Grandy (HP) in 1987 and the updated version i.e FURPS+ in 2000 by IBM Rational Software
- Categorized into **Functional (F)** and **Non-Functional (URPS)**

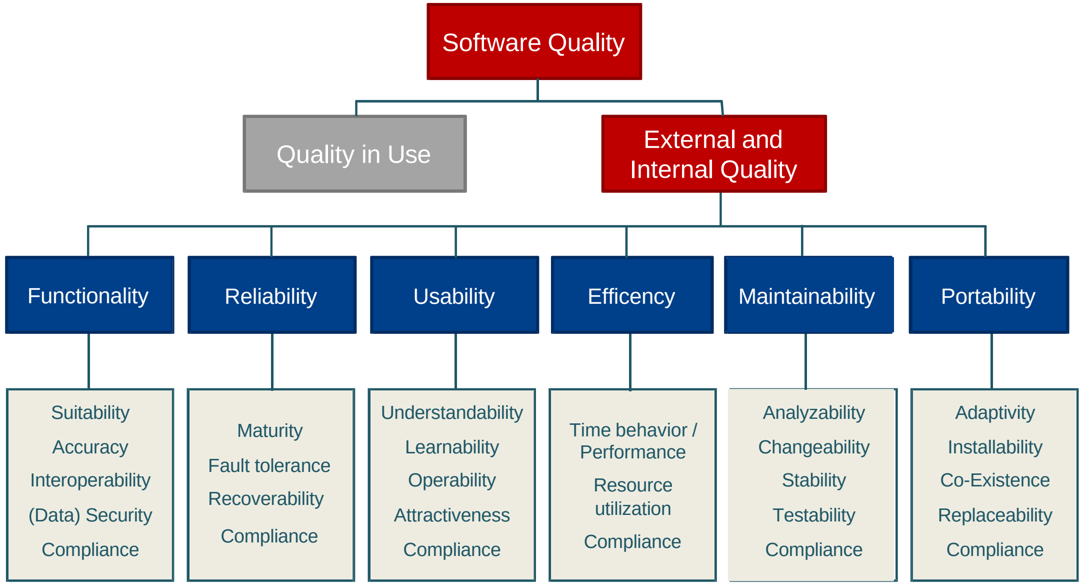


Quality

- Functionality
  - Feature set
  - Capabilities
  - Generality
  - Security
- Usability
  - Human factors
  - Aesthetics
  - Consistency
  - Documentation
- Reliability
  - Frequ./severity of fail.
  - Recoverability
  - Predictability
  - Accuracy
  - Mean time to failure
- Performance
  - Speed
  - Efficiency
  - Rsc. consumption
  - Thruput
  - Response time
- Supportabilit y
  - Testability
  - Extensibility
  - Adaptability
  - Maintainability
  - Compatibility
  - Configurability
  - Serviceability
  - Installability
  - Localizability

Software Quality Model – **ISO 9126 Model**

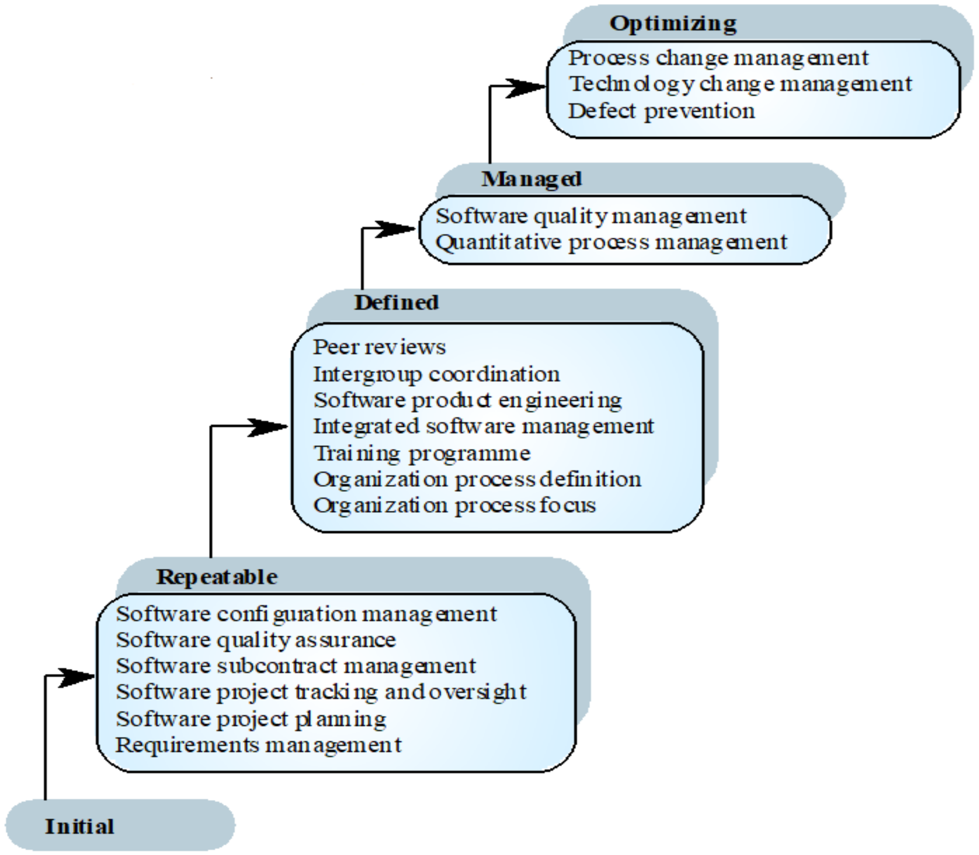
- Is the **international** level of software quality model
- Claims to be a comprehensive list
- Gives guidance on refining the factors and defines some possible sub-attributes



Source:
ISO/IEC 9126: Evaluation of software products, quality characteristics and guidance on usage

Software Quality Model – **CMMI Model**

- Is a methodology used to develop and refine an organization's software development process.
- Consists of 5 maturity levels
  1. **Initial.** The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort and heroics.
  2. **Repeatable.** Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.
  3. **Defined.** The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization. All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software.
  4. **Managed.** Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.
  5. **Optimizing.** Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.



Optimizing

Process change management
Technology change management
Defect prevention

Managed

Software quality management
Quantitative process management

Defined

Peer reviews
Intergroup coordination
Software product engineering
Integrated software management
Training programme
Organization process definition
Organization process focus

Repeatable

Software configuration management
Software quality assurance
Software subcontract management
Software project tracking and oversight
Software project planning
Requirements management

Initial

The need for comprehensive definition of requirements

- Many cases of low customer satisfactions are due to the product that does not fulfilled the basic customers requirements (requirements of correctness, poor performance, maintenance, reliability, software reuse, training, etc)
- **Why??**
  - Lack of defined requirements pertaining to these aspects of software functionality
- **Thus..**
  - There is a need for a **comprehensive definition of requirements** that will cover all aspects of software use throughout all stages of SDLC

Classification of software requirements into quality factors

- Quality models define the broad spectrum of software requirements.

How?

- Refer to each quality factor of the model,
- Examine it and accordingly incorporate the respective requirements in the requirements documents

Outcome..

Quality Requirements!

Criteria of Quality Requirements

- Consistent
- Unambiguous
- Understandability
- Traceability
- Testability/Verifiable
- Clear (Concise, Simple, Precise)
- Feasible (Realistic, possible)

- Correct
- Necessary
- Consistent
- Non-redundant
- Complete
- Independent

Software Quality Requirements

Who interested in definitions of quality requirements?

- All Stakeholders! - Clients/Customers, Developers, Mangers, System Analys, Team Leaders, Tester, etc
- Analysis of various quality factors from various quality models indicates how each role in SDLC can add quality requirements that represents his own interest.
- Example
  - Reusability requirements
    - Project manager anticipates that some of the modules can be used in other similar project
  - Verifiability requirements
    - Incorporated by the development team as the verifiability requirements are meant to improve design reviews and software tests carried out during software development. The aim is to save development resources.

Quality factors and Quality Requirements - Example

- **Correctness** requirements are defined in a list of the software **system's required outputs**, such as a query display of a customer's balance in the sales accounting information system.
  - The output mission e.g. sales invoice printout
  - The required accuracy of those outputs that can be adversely affected by inaccurate data or calculations
  - The completeness of the output information
  - The up-to-date information
  - The availability of information
  - The standards for coding and documenting the software system

Quality factors and Quality Requirements - Example

Correctness

A club membership information system include:

- ▶The output mission : a defined list of 8 types of reports, standard letters, and queries
- ▶The required accuracy of those outputs : the probability for a non-accurate output will not exceed 1%
- ▶The completeness of the output information : the probability of missing data will not exceed 1%
- ▶The up-to-date information : not more than 2 working days
- ▶The availability of information : reaction time for queries will be less than 2 seconds and less than 4 hours for reports
- ▶The standards for coding and documenting the software system : comply with client’s guidelines and standards defined by the project management

Quality factors and Quality Requirements - Example

Reliability

Reliability requirements deal with failures to provide service.

Example :

The failure frequency of a heart-monitoring unit is required to be less than one in 20 years. Its heart attack detection function is required to have a failure rate of less than 1 per million cases

Quality factors and Quality Requirements - Example

Efficiency

Efficiency requirements deal with hardware resources needed to perform all the functions of the software system in conformance to all other requirements. Mainly concerned with CPU speed, storage capacity, network

Example : An outdoor meteorological unit, equipped with a 1000 milli-ampere hour cell, should be capable of supplying the power requirements of the unit for at least 30 days. The system performs measurements once per hour, logs the results, and transmits the results once a day to the meteorological center by means of wireless communication

Quality factors and Quality Requirements - Example

Integrity

Integrity requirements deal with the software system security; to prevent access to unauthorized persons

Example : An engineering department operates a Geographic Information System. It is planning to allow citizens to access its GIS files through the Internet. The software requirements include the possibility of viewing and copying but not inserting changes in the maps. Access will be denied to plans in progress and to those maps defined as limited access documents

Quality factors and Quality Requirements - Example

Efficiency

Efficiency requirements deal with the scope of staff resources needed to train a new employee to operate the software system.

Example : The software **Efficiency** requirements document for a new help desk system has the following specifications:

- ➢A staff member should be able to handle at least 60 service call per day
- ➢Training a new employee will take no more than 2 days after which the trainee will be able to handle 45 service calls per day

Quality factors and Quality Requirements - Example

Maintainability

Maintainability requirements determine the efforts needed by users and maintenance personnel to identify the reasons for software failures, to correct and to verify the success of corrections. Refers to the modular structure of software, the internal program documentation, and the programmer's manual

Examples:

- The size of a software module will not exceed 30 statements
- The programming will adhere to the company coding standards and guidelines

Quality factors and Quality Requirements - Example

Flexibility

The capabilities and efforts required to support adaptive maintenance activities including the resources required to adapt a software package to a variety of customers

Examples:

- The software should be suitable for teachers of all subjects and all school levels
- The teachers should be able to create new types of reports according to their requirements

Quality factors and Quality Requirements - Example

Testability

Testability requirements deal with testing of an information system including automatic diagnostics to detect faults, log files

Example:

An industrial computerized control unit is programmed to calculate various measures of production status, report the performance level of the machinery, and operate a warning signal in predefined situations. One testability requirement demanded was to develop a set of standard test data with known system expected correct reactions in each stage to be run every morning, before production begins, to check if the computerized unit reacts properly

Quality factors and Quality Requirements - Example

Portability

Portability requirements concern with the adaptation of a software system to other environments consisting of different hardware, different O/S

Example:

A software package designed and programmed to operate in a Windows 2000 environment is required to allow transfer to Linux and Windows NT environments

Quality factors and Quality Requirements - Example

Interoperability

Interoperability requirements focus on creating interfaces with other software systems or other firmware.

Example:

The firmware of a medical lab's equipment is required to process its results according to a standard data structure that can then serve as input for a number of standard lab information systems

See you NEXT ⇨ Class..