Benghazi University Faculty of Information Technology

Software Engineering Department

Software Quality

Part3

(Quality Attributes & Models)

Instructor: Salah Abdelsattar

Software Quality attributes

- Quality attributes are the measurable or observable characteristics of products and processes.

- Quality attributes are ***used early in the development process*** to identify user quality requirements. Each system has specific and unique quality needs, which are a function of the purpose of the application.

- Quality attributes can be classified into internal & external quality attributes:
  - **External**: Derived from the relationship between the environment and the system(or the process). (To derive, the system or process must run)
  - **Internal**: Derived immediately from the product or process description (To derive, it is sufficient to have the description)

- **Underlying assumption**: internal quality leads to external quality

Internal & external Software Quality attributes

- Internal quality concerns the developers/maintainers of the software.
  - Software with a high internal quality is easy to change, easy to add new features, and easy to test.
  - Software with a low internal quality is hard to understand, difficult to change, and troublesome to extend.
- External quality concerns the users of the software. They are a measure of how the system as a whole meets the requirements of its stakeholders.
  - Does the system provide the functionality required?
  - Is the interface clear, easy to use, and consistent?
  - Does the software provide the expected business value?
- The difference between internal and external attributes isn’t completely clear-cut because at some level internal attributes affect external ones.

Examples of internal & external Software Quality attributes

Internal Quality attributes:

- Maintainability,
- Flexibility,
- Portability,
- Re-usability,
- Readability,
- Testability
- Understandability.

External Quality attributes:

- Correctness,
- Usability,
- Efficiency,
- Reliability,
- Integrity,
- Adaptability,
- Accuracy
- Robustness.

Definitions and measures of Some software Quality Attributes

Maintainability - is easy to change and adapt to new requirements

- measured in terms of change logs (time and effort required to add a new feature) and impact analysis (#lines affected by a new feature)

Correctness - lack of bugs and defects

- measured in terms of defect rate (# bugs per line of code)

Usability - is sufficiently convenient for the intended users

- measured in terms of user satisfaction (% of users happy with interface and ease of use)

Reliability - does not fail or crash often

- measured in terms of failure rate (#failures per hour)

Capability - does all that is required

- measured in terms of requirements coverage of required operations implemented)

Some software Quality Attributes definitions and measures …

Performance - is fast and small enough

- measured in terms of speed and space usage (seconds of CPU time, Mb of memory, etc.)

Installability - is convenient and fast to install

- measured in terms of user satisfaction (#install problems reported per installation)

Documentation - is well documented

- measured in terms of user satisfaction (% of users happy with documentation)

Availability - is easy to access and available when needed

- measured in terms of user satisfaction (% of users reporting access problems)

Software Quality models

- A quality model is a set of selected **quality** characteristics (attributes) and sub- characteristics together with the assigned measures.
- The **quality model** provides the basis for specifying **quality**

requirements and evaluating **quality**.

- Well known software quality models are:
  - McCall Model
  - Boehm Model
  - FURPS Model
  - Dromey Model
  - ISO/IEC 9126 Model

McCall Software Quality Model

Classifies 11 quality factors (attributes) into three categories

- **Product Revision factors**
  - Related to the ability of software to undergo changes
- **Product Transition factors**
  - Related to the software adaptability to new environments
- **Product Operations factors**
  - Related to the daily operation of the software

McCall Software Quality Model

Factor	Criteria	Description
Product Revision	Maintainability	Can I fix it?
	Flexibility	Can I change it?
	Testability	Can I test it?
Product Transition	Portability	Will I be able to use it on another machine?
	Reusability	Will I be able to reuse some of the software in other application?
	Interoperability	Will I be able to interface it with another system?
Product Operation	Correctness	Does it do what I want?
	Reliability	Does it do it accurately all the time?
	Efficiency	Will it run as well as it can?
	Integrity	Is it secure?
	Usability	Is it easy to use?

McCall Software Quality Model

1) **Product Revision (changing it)**

- ***Maintainability*** *(can I fix it?)*

The effort required to locate and fix an error in a program. The system should be easy to keep up for its intended use. Changes for improving operational efficiency should be easy to implement. Failed operations should be easy to restore to satisfactory condition.

- ***Flexibility*** *(can I change it?)*

The effort required to modify an operational program. Change and enhancement of the system should be easily implementable.

Testability *(can I test it?)*

The effort required to test a program to ensure that it performs its intended function. The ability of the system to produce quality product units should be easily testable. Useful messages should be generated for testing and debugging purposes.

McCall Software Quality Model

2) Product Transition (modifying it to work in a different environment)

- ***Interoperability*** *(Will I be able to interface it with another system?)*

The effort required to couple one system to another.

- ***Portability*** *(Will I be able to use it on another machine?)*

The effort required to transfer the program from one hardware and/or software system environment to another. The system should be portable among people and among machines. Attainment of the other quality characteristics greatly facilitates portability.

- ***Reusability*** *(Will I be able to reuse some of the software?)*

The extent to which a program (or part of a program) can be reused in other applications-related to the packaging and scope of the functions that the program performs.

McCall Software Quality Model

3) Product Operations (using it)

- ***Correctness*** *(Does it do what I want?)*

The extent to which a program satisfies its specification and fulfills the customer’s mission objectives. The extent to which software is free from design defects and from coding defects. That is fault-free.

Reliability *(Does it do it accurately all of the time?)*

The extent to which a program can be expected to perform its intended function with required precisions under stated conditions for a stated period of time.

- ***Efficiency*** *(Will it run on my hardware as well as it can?)*

The extent to which a software performs its function. With a minimum consumption of computing resources. It should not use any hardware components unnecessarily.

- ***Integrity*** *(Is it secure?)*

The extent to which access to software or data by unauthorized persons can be controlled.

- ***Usability*** *(Is it designed for the use?)*

The effort required to learn, operate, prepare input, and interpret output of a program.