



Software Quality Factors



The need for a comprehensive software quality requirements

- There are some characteristic common :
 - All the software projects satisfactory fulfilled the basic requirements for correct calculations
 - All the software projects suffered from poor performance in important areas such as maintenance, reliability, software reuse, or training.
 - The cause for the poor performance of the developed software projects in these areas was the lack of predefined requirements to cover these important aspects of the software's functionality.



The need for a comprehensive definition of requirements (2)

- **There is a need for a comprehensive definition of requirements that will cover all attributes of software and aspects of the usability aspects, reusability aspects, maintainability aspects, and so forth in order to assure the full satisfaction of the users.**



Classification of software requirements into software quality factors

- The classic model of software quality factors suggest by :
 - McCall (consist of 11 factors, 1977)
 - Deutsch and Willis (consist of 12 to 15 factors, 1988)
 - Evans and Marciniak (1987)



McCall's Factor Model

- Classifies all software requirement into 11 software quality factors, grouped into three categories :
 1. **Product operation factors** : Correctness, Reliability, Efficiency, Integrity, Usability.
 2. **Product revision factors** : Maintainability, Flexibility, Testability.
 3. **Product transition factors** : Portability, Reusability, Interoperability.



Product Operation Factors



Product Operation : Correctness

- Correctness requirements are defined in a list of the software system's required outputs.
- Output specifications are usually multidimensional; some common dimensions include :
 - The output mission
 - The required accuracy of output
 - The completeness of the output information
 - The up-to-dateness of the information
 - The availability of the information
 - The standards for coding and documenting the software system.



Product Operation - Reliability

- Reliability requirements deal with failures to provide service.
- Determine the maximum allowed software system failures rate, and can refer to the entire system or to one or more of its separate functions.
- Examples :
 1. The failures frequency of a heart-monitoring unit that will operate in a hospital's intensive care ward 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 one per million cases.
 2. One requirement of the new software system to be installed in the main branch of Independence Bank, which operates 120 branches, is that it will not fail, on average, more than 10 minutes per month during the bank's office hours.



Product Operation - Efficiency

- **Efficiency** requirements deal with the hardware resources needed to perform all the functions of the software system in conformance to all other requirements.
- **Examples :**
 1. A chain stores is considering two alternative bids for a software system.



Product Operation - Integrity

- **Integrity** requirements deal with the software system security, that is requirements to present access to an authorize person,

[to distinguish between the majority of personnel allowed to see the information (“read permit”) and a limited group who will be allowed to add and change data (“write permit”), and so forth.]

- Example :
GIS SW allowed citizens access to its GIS through Internet only for viewing and copying data but not to insert changes.



Product Operation - Usability

- **Usability** requirements deal with the scope of staff resources needed to train a new employee and to operate the software system.

- Example :
The software usability requirements document for the new help desk system initiated by a home appliance service company lists the following specifications :
 - a. A staff member should be able to handle at least 60 service calls a day
 - b. Training a new employee will take no more than two days, immediately at the end of which the trainee will be able to handle 45 service calls a day.



Product Revision Factors



Product Revision software Quality Factors

- According to the McCall model of software quality factors, three quality factors comprise the product revision category. These factors deal with those requirements that affect the complete range of software maintenance activities :
 - Corrective maintenance
 - Adaptive maintenance
 - Perfective maintenance



Product Revision - Maintainability

- **Maintainability** requirements determine the efforts that will be needed by users and maintenance personnel to identify the reasons for software failures, to correct the failures, and to verify the success of the corrections.
- This factor's requirements determine refer to the modular structure of software, the internal program documentation, and the programmer's manual, among other items.
- Example typical maintainability requirements :
 - a. The size of a software module will not exceed 30 statements.
 - b. The programming will adhere to the company coding standards and guidelines.



Product Revision - Flexibility

- The capabilities and efforts required to support adaptive maintenance activities are covered by the **flexibility** requirements.
- These include the resources (in man-days) required to adapt a software package to a variety of customers of the same trade, of various extents of activities, of different ranges of products, and so on.
- This factor's requirements also support perfective maintenance activities.



Product Revision - Testability

- **Testability** requirements deal with the testing of an information system as well as with its operation.
- Testability requirements for the ease of testing are related to special features in the programs that help the tester, for instance by providing predefined intermediate results and log files.



Product Transition Factors



Product Transition - Portability

- **Portability** requirements tend to the adaptation of software system to other environments consisting of different hardware, different operating systems, and so forth.



Product Transition - Reusability

- **Reusability** requirements deal with the use of software modules originally designed for one project currently being developed.



Product Transition - Interoperability

- **Interoperability** requirements focus on creating interfaces with other software systems or with other equipment firmware.
- **Interoperability** requirements can specify the name of the software or firmware for which interface is required.



Alternative Models of Software Quality Factors

- Formal comparison of the alternative models (Evans M 1987, Deutsch & Willis 1988)
 - Comparison of the factors models-content analysis (verifiability, expandability, safety, manageability, survivability)
 - Structure of the alternative factor models



Who is interested in the definition of quality requirements?

- Some quality factors not included in the typical client's requirements document may, in many cases, interest the developer.
- The following lists of quality factors usually interest the developer whereas they may raise very little interest on the part of the client :
 - Portability
 - Reusability
 - Verifiability



Requirements Documents

A project will be carried out to according to two requirements documents :

- The client's requirements document
- The developer's additional requirements document



Software Compliance with Quality Factors

- The software's product compliance to the requirements belonging to the various quality factors is measured by software quality metrics, measures that quantify the degree of compliance.
- In order to allow for valid measurements of compliance, sub-factors have been defined for those quality factors that represent a wide range of attributes and aspects of software use.

Factors and sub-factors

Factor model	Software Quality Factors	Sub-factors
McCall's model: Product Operation category	Correctness	Accuracy Completeness Up-to-dateness Availability (response time) Coding and documentation guidelines Compliance (consistency)
	Reliability	System reliability Application reliability Computational failure recovery Hardware failure recovery
	Efficiency	Efficiency of processing Efficiency of storage Efficiency of communication Efficiency of power usage (for portable unit)
	Integrity	Access control Access audit
	Usability	Operability Training
McCall's model: Product Revision category	Maintainability	Simplicity Modularity Self-descriptiveness Coding and documentation guidelines Compliance (consistency) Document Accessibility

Factors and sub-factors (cont.)

McCall's model:
Product Revision category

Flexibility

Modularity
Generality
Simplicity
Self-descriptiveness

Testability

User testability
Failure maintenance testability
Traceability

McCall's model:
Product Transition category

Portability

Software system independence
Modularity
Self-descriptive
Generality
Simplicity

Reusability

Modularity
Document Accessibility
Software system independence
Application independence
Self-descriptive
Generality
Simplicity

Interoperability

Commonality
System compatibility
Software system independence
Modularity

Factors of the alternative
models

Verifiability

Coding and documentation guidelines
Compliance (consistency)
Document Accessibility
Traceability
Modularity



Factors and sub-factors (cont.)

Factors of the alternative models

Expandibility

Extensibility
Modularity
Generality
Simplicity
Self-descriptiveness

Safety

Avoidance of hazardous operating situations
Unsafe conditions alarm reliability

Manageability

Completeness and ease of support of infrastructure services for software modification in the development process
Completeness and ease of support of infrastructure services for software modification in the maintenance process

Survivability

System reliability
Application reliability
Computational failure recovery
Hardware failure recovery



Summary

1. The need for a comprehensive requirements documents and their contents.
2. The structure (categories and factors) of McCall's classic factor model.
3. The additional factors suggested by alternatives factor models.
4. Those interested in defining software quality requirements.



Questions ?

Thank you