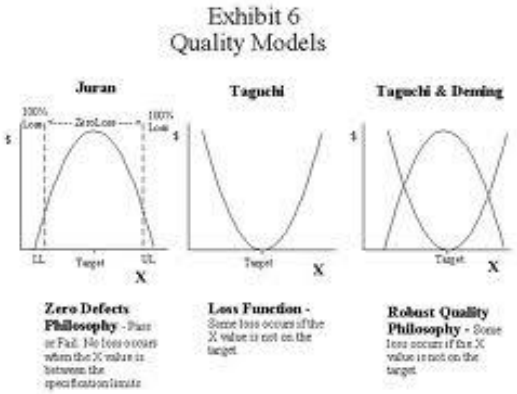
SOFTWARE QUALITY MODELS

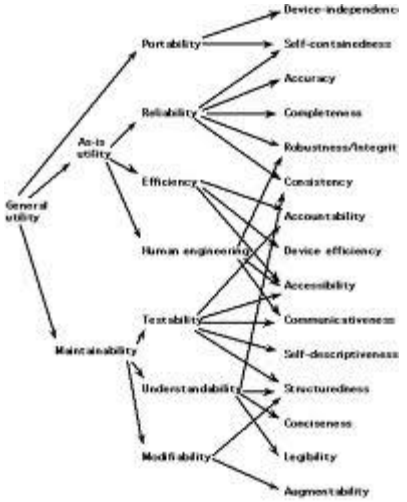
Software Quality Models



- Models about software quality in terms of
  - The factors that affect software quality
    - Attributes directly indicate the quality of he system, e.g. reliability, correctness, user friendliness, etc.
    - Attributes indirectly related to quality, e.g. internal complexity, etc.
  - The interrelations between the factors
    - Causality models:
      - Causal relationship, in terms of stereo-type relations
    - Quantitative models:
      - Quantitative relations expressed as numerical functions,
      - Numerical values of the basic attributes are given, e.g. through using metrics/measurements
      - The overall quality in an attribute on a more high level of abstraction is calculated numerically

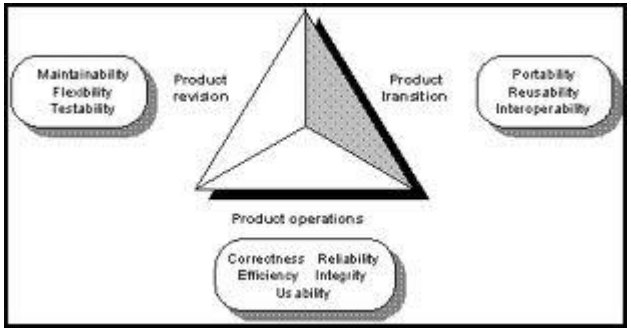
Hierarchical Models of SQ

- A set of quality related properties organised into a hierarchical structure
  - E.g. decompose quality into a number of quality attributes and attributes into a number of factors, and then a number of measures, etc.
  - Positive causal relationship is represented
- Examples:
  - McCall model (1977)
  - Boehm model (1978)
  - ISO model (1992)
  - Bansiya-Davis model of OO designs (2002), etc.



GE Model (McCall Model 1977&1980) (General Electric Model)

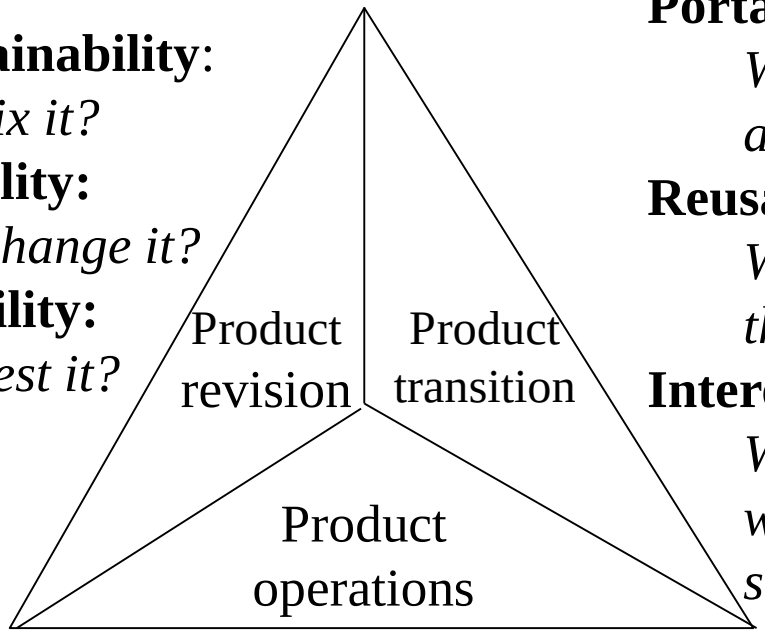
- McCall's QUALITY Model (1977)
  - Shows relationships between external QUALITY Factors such as correctness, reliability, usability, testability, etc. and Product QUALITY Criteria such as traceability, completeness, error tolerance, etc.
  - Areas Addressed are
  - Product Operation
  - Product Revision
  - Product Transition



- PROBLEM: Difficult to measure these QUALITY Factors
  - No Standards, No Methods, No Tools

McCall's Model

Maintainability:
Can I fix it?
Flexibility:
Can I change it?
Testability:
Can I test it?



Portability:
Will I be able to use it on another machine?
Reusability:
Will I be able to reuse some of the software?
Interoperatability:
Will I be able to interface it with another machine / software?

Correctness: *Does it do what I want?*
Reliability: *Does it do it accurately all the time?*
Efficiency: *Will it run on my machine as well as it can?*
Integrity: *Is it secure?*
Usability: *Can I run it?*

Boehm's QUALITY Model (1986)

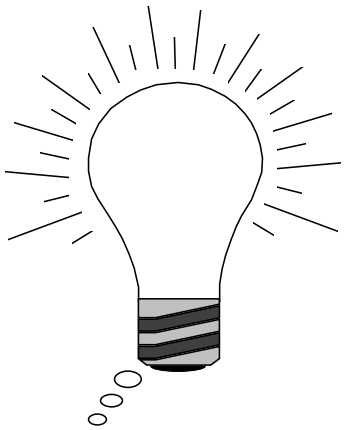
- Asserts that QUALITY software satisfies the needs of the users, designers, testers, and maintainers
- Relates software's general utility to maintainability, reliability, testability, etc., to device independence, completeness, accessibility, etc.
- Very difficult to apply in practice, but easy to understand and learn
- Parts of it can be supported by tools; therefore, useful in a restricted sense

(Boehm, 1978)

Portability

Efficiency

Testability



Reliability

Understandability

Usability

Modifiability