BLEKINGE INSTITUTE OF TECHNOLOGY

BTH

Software Metrics (PA1407)

Lecture 5

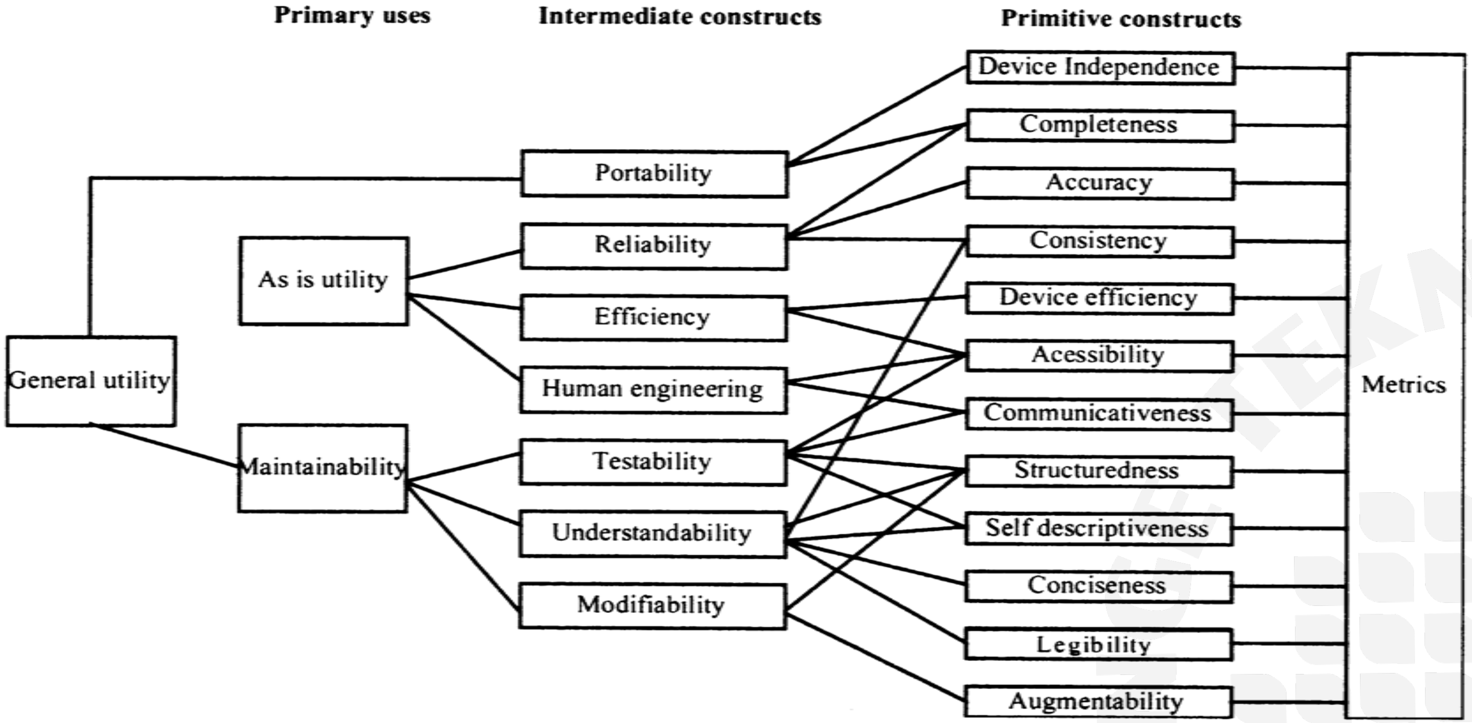
Measuring External Product Attributes

Software Quality

- Views
  - Fitness for purpose
  - Conformance to specification
  - Degree of excellence
  - Timeliness
- ISO 9126 definition of quality:
  - *The totality of features and characteristics of a software product that bear on its ability to satisfy stated or implied needs*

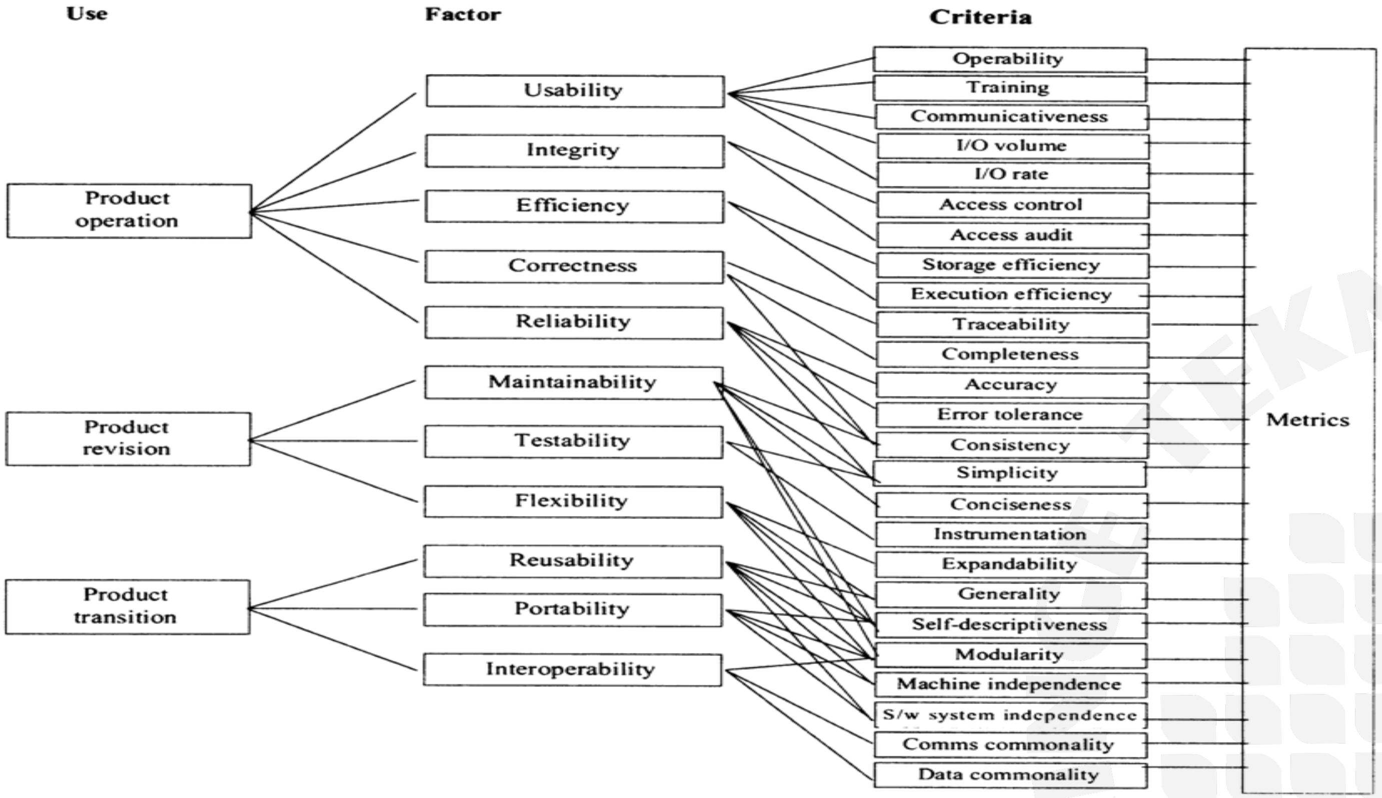
Modeling software quality

Boehm software quality model



Modeling software quality

McCall software quality model



Quality standards

- ISO 9126-1 (1992) : Six characteristics
  - Functionality
  - Reliability
  - Efficiency
  - Usability
  - Maintainability
  - Portability
- ISO/IEC 25010 (2011) replaced ISO/IEC 9126: Eight characteristics
  - Functional suitability
  - Performance efficiency
  - Compatibility
  - Usability
  - Reliability
  - Security
  - Maintainability
  - Portability

Defects-based quality measures

- Defect density measures
  - Defect density = No. of defects / product size

- Other
  - System spoilage = $\frac{\text{time to fix post-release defects}}{\text{total system development time}}$

- Next, we will focus on only two quality characteristics: Usability and Maintainability.
- Other quality characteristics have their own attributes and corresponding measures.

Usability measures

- The ISO/IEC 25010 defines usability as

  "Usability is the degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."

- External view
  - Effectiveness measures indicate the degree to which users can correctly complete tasks
    - E.g. Counts or percentage of completed tasks and errors made, user recall etc.
  - Efficiency measures generally involve time required to complete tasks.
    - E.g. Input rate, mental effort required to complete a task.
  - Satisfaction measures indicate subjective notions of the quality of interaction with a system.
    - User's experiences via survey, pulse rates and facial expressions etc.

Usability measures

- Internal view of usability
  - Collecting early data on usability is difficult, before system is developed
  - One way is to look for internal attributes that we think lead to good usability
    - Well-structured manuals
    - Good use of menus and graphics
    - Informative error messages
    - Help functions
    - Consistent interfaces
  - We can measure these internal attributes. However, we cannot define them to be measures of usability.

Maintainability measures

- Types of maintenance
  - Corrective
  - Adaptive
  - Preventive
  - Perfective
- External view of maintainability
  - Mean Time TO Repair (MTTR)
    - Problem recognition time
    - Administrative delay time
    - Maintenance tools collection time
    - Problem analysis time
    - Change specification time
    - Change time (including testing and review)

Maintainability measures

- External view of maintainability : Other measures
  - Ratio of total change implementation time to total number of changes implemented
  - Number of unresolved problems
  - Time spent on unresolved problems
  - Percentage of changes that introduce new faults
  - Number of modules modified to implement a change.
- All of these measures paint a picture of the degree of maintenance activity and the effectiveness of the maintenance process.

Maintainability measures

- Internal attributes affecting maintainability
  - A number of the "complexity" measures, discussed in previous lecture, have been correlated significantly with levels of maintenance effort.
  - We need to remember that correlation with a characteristic does not make something a measure of that characteristic
    - So we continue to separate the structural measures from maintainability measures
  - Nevertheless, some of the structural measures can be used for "risk avoidance" with respect to maintainability.
  - These internal measures must be gathered in combination with external maintainability measures.
  - On the basis of accumulated evidence, we may, for example, identify a particular module having measurably poor structure.

Acknowledgement

- Lecture notes are prepared from following sources:
  - T1: Software Metrics - A Rigorous & Practical Approach, 2nd edition, Authors: N. E. Fenton, S. L. Pfleeger, Publishers: International Thomson Computer Press, 1996, ISBN: 1-85032-275-9.