



Software Measurement

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

Software Measurement

- The goal of software measurement is to make judgements about software quality
- Classification of software metrics
 - *Process metrics*: support management (e.g., time for a particular task)
 - *Product metrics*: are associated with the software itself (e.g., LOC)



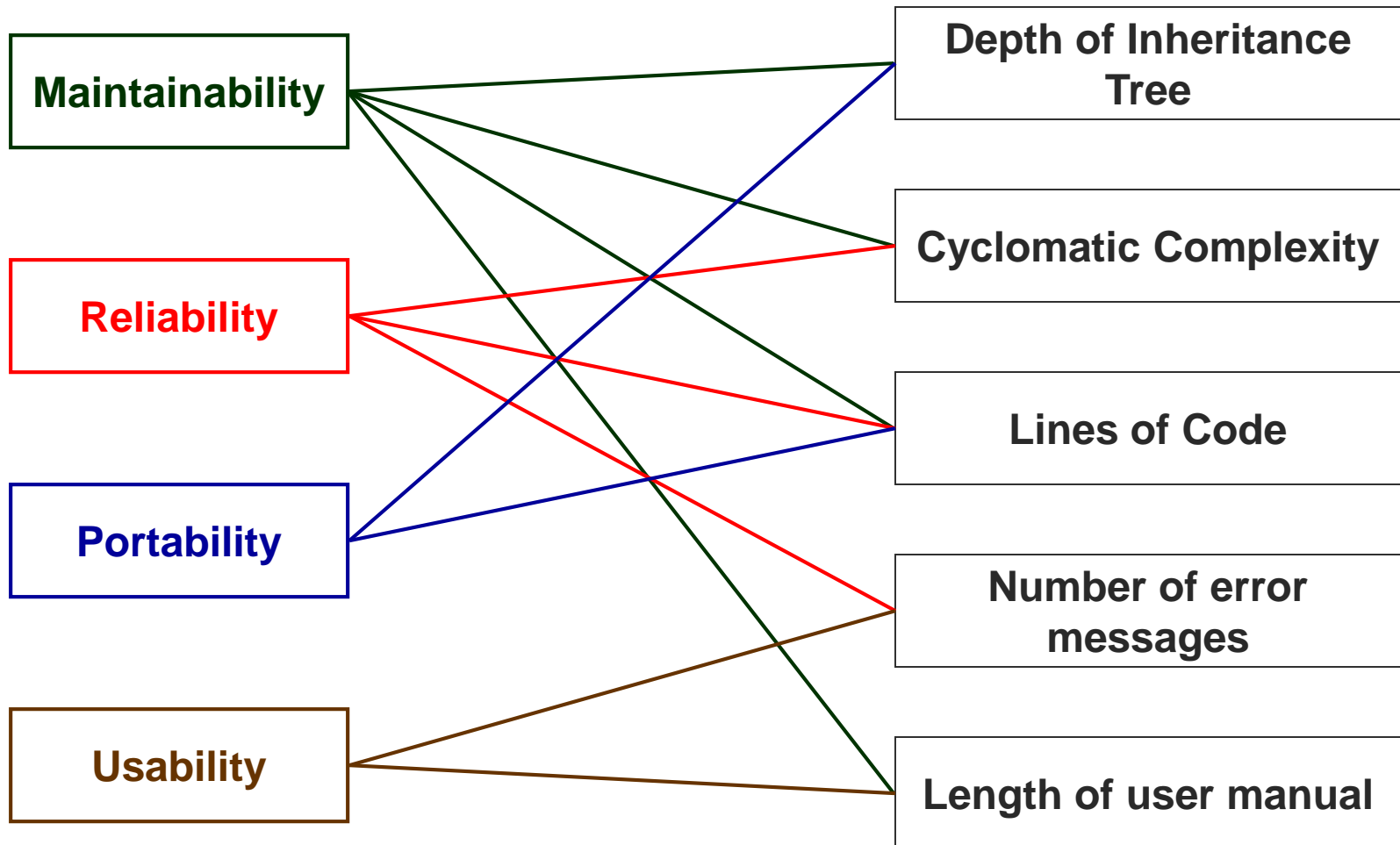
[Challenge of Software Metrics]

- It is difficult to directly measure an external quality attribute
 - Software metrics usually quantify only internal attributes (or factors)
- Examples of quality attributes
 - Maintainability
 - Usability
 - Reliability

[Quality Models]

- They relate internal attributes to external ones
 - Internal attributes are easier to be measured
- Relationships in quality models must be clear and valid, otherwise the model is useless

Example of Quality Model



[Validity of Quality Models]

- At least three conditions must be verified in quality models
 - An internal attribute must be precisely measured
 - Internal attributes must be related to external attributes
 - Relationships must be understood and validated





Software Component Analysis

[Measurement Process]

- A measurement process usually have five main activities:
 1. Choose measurement to be made
 2. Select the components to be assessed
 3. Measure internal attributes of components
 4. Identify anomalous measurements
 5. Analyze anomalous components

[Product Measurement Process]

**Choose
measurements**



**Select
components**



**Measure
components**



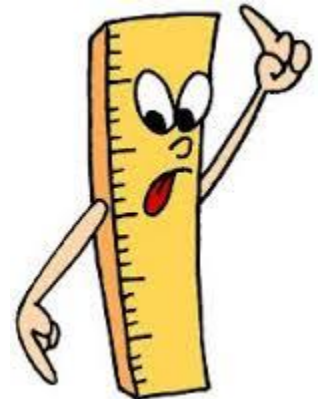
**Identify
anomalous
measures**



**Analyze
anomalous
components**

[1 - Choosing Metrics]

- In this activity, you may want to use the GQM method
 - *Goal-Question-Metric*
- Questions are formulated based on a more abstract goal
- Metrics are chosen to answer each question



[The GQM Method]

■ Goals

- They define what the organization wants to improve (e.g., productivity)

■ Questions

- They refine each goal to a more quantifiable way (e.g., how can we increase the amount of code we produce?)

■ Metrics

- They indicate the metrics required to answer a question (e.g., current LOC per developer)

[2 - Selecting Components]

- It might not be necessary (or desirable) to measure the whole system
- Selection strategies
 - To choose a subset of representative components
 - To choose those components which seem to be critical in the software system

[3 - Measuring Components]

- The selected components are measured
 - Measures are related to quality attributes (e.g., by means of a quality model or GQM)
- Measurement tools can be integrated with the development environment



[4 - Identifying Anomalous Measures]

- After measures have been collected, you can compare them
 - Historical data may be required in this step
- The analysis should focus on anomalous values
 - That is, values too high or too low for a particular metric



[5 - Analyzing Components]

- If a component has anomalous measures, it has to be further analyzed
 - An manual inspection process decides if a problem does exist in that component
- An uncommon value does not always mean a problem
 - Analysis of measurement is not easy
 - It may require statistic methods



[Example (Excel File)]

- Measuring several versions
 - Product metrics collected in the Health Watcher system
- Metrics
 - VS: Vocabulary Size
 - CBO: Coupling Between Object Classes
 - LCOM: Lack of Cohesion over Methods
 - DIT: Depth of Inheritance Tree
 - NOC: Number of Children
 - LOC: Lines of Code, etc.

[Bibliography]

- Ian Sommerville. **Software Engineering**, 10th Edition. Pearson Education, 2016.
 - Section 24.5: Software Measurement