The University of Western Australia
School of Computer Science and Software Engineering

CITS5502 Software Processes

Lecture 4
Revision of Measurement Theory
Key concepts

- **What is a Metric?**
  - Measurement of some abstract property of a system or process
  - Relationship of *Precision*, *Accuracy*, and *Relevance*
  - Mathematical axioms for a metric
  - Desirable properties of a metric

- **Purpose of Metrics**
  - Prediction; Control; Improvement; Optimization

- **Aspects of definition of a metric**
  - Measurement leads to improvement
  - Non-measurement leads to deterioration
  - Metrics must be aligned with organizational objectives
Key concepts (cont.)

- Type of measurement and use – statistical methods

- Validation of metrics
  - Mathematically, Qualitative, Provides information

- Simple taxonomy of software metrics
  - Quality; Product; Process; Organization and Drivers
What is a Metric?

- A metric is a formal measurement designed to associate a number with some characteristic of the things you want to assess.

- It is not always straightforward to define a metric on what you want to measure, e.g., *How would you measure the size of a system?*
Assessing a useful metric

Mathematically valid w.r.t. a set of axioms
- Validity of set of axioms
- Correlates with opinion of group of experts
- Complies with standards for metrics
- Complies with a set of desirable properties

Qualitatively valid w.r.t. some criteria

Provides useful information
- Accurate prediction
- Warning of loss of control
- Improvement of process
- Optimization given data on constraints
Examples

- Suppose that we want to estimate how long (in months) it would take to do a project, we can use the formula below to estimate the error:

\[
\text{Error} = \frac{|\text{Estimate} - \text{Actual}|}{\text{Actual}}
\]

This is the relative error of our estimate.

- If we have \( n \) projects, then we can compute the mean magnitude of relative error (MMRE):

\[
\text{MMRE} = \frac{1}{n} \sum_{i=1}^{n} \text{Error}_i
\]
## Types of Measurement

<table>
<thead>
<tr>
<th>Scale Name</th>
<th>Operations</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal</td>
<td>Comparison</td>
<td>Companies, Languages</td>
</tr>
<tr>
<td>Ordinal</td>
<td>Ranking</td>
<td>Preferences, Relative sizes</td>
</tr>
<tr>
<td>Interval</td>
<td>Addition, Subtraction</td>
<td>Time, Temperature</td>
</tr>
<tr>
<td>Ratio</td>
<td>Multiplication, Division</td>
<td>Length, Size</td>
</tr>
<tr>
<td>Absolute</td>
<td></td>
<td>Census</td>
</tr>
</tbody>
</table>

Nominal and Ordinal data items may (for convenience) have numeric labels (e.g., Aboriginal=1, Caucasian=2, Mongoloid=3, Negro=4) but this does not imply that one is twice as great as another!
Examination of Data

- For statistical purposes, the distribution of a data item may be inspected by means of
  - a histogram with a mode for nominal data
  - a histogram with a median for ordinal data
  - a continuous distribution with a mean for interval, ratio, or absolute data

- Two data items of the same type may be compared by means of
  - a contingency table for nominal or ordinal data
  - a scattergram for interval, ratio, or absolute data
Combining data items of different types produces an item with the characteristics of the lower type.

- For example, **Size** (in lines of code [ratio]) multiplied by **Complexity** (judged on a 1 to 5 scale [ordinal]) of a piece of computer code produces a data item that looks like a ratio number but only has the precision of an ordinal number.
Desirable Properties for Software Metrics

- **Conforming to the needs of people**
  1. **Understandable** – the measurement mechanism is easily understood by all affected parties
  2. **Non-threatening** – focused on the artefact and not people

- **Conforming to measurement theory**
  3. **Objective** – is independent of anyone’s opinion
  4. **Reproducible** – can be consistently repeated
  5. **Standardized** – uses a mathematically appropriate scale
  6. **Formally valid** – conforms to sets of mathematical axioms
  7. **Informally valid** – is clearly related to the feature being measured – it monotonically increases as the feature rises
  8. **Precise** – discriminates between levels of the feature measured (i.e., it is sensitive to changes at the required level of granularity)
Desirable Properties for Software Metrics (cont.)

- Conforming to measurement theory
  9. Robust – is not easily manipulated nor sensitive to extraneous factors
  10. Comparable – is highly correlated with other metrics purporting to measure the same feature
  11. Useful – provides information which reduces uncertainty

- Conforming to the needs of management
  12. Sustainable over time – is likely to be valid in the future so that trend forecasts based on the metric will be useful
  13. Scalable – can be translated into metrics for macro or micro parts of the product or process
  14. Timely – can be obtained in time for action to be taken on its message
Desirable Properties for Software Metrics (cont.)

- Conforming to the needs of management
  15. **Effective** – clearly identifies significant areas for improvement
  16. **Efficient** – does not consume significant resources (preferably a by-product of other activities)
  17. **Relevant** – supports the goals of the organization
Recommended Reading

- Royce: Appendix C “Change Metrics”
- Wysocki: Appendix J “Project Performance Reporting”
- Pressman: Sections on “FURPS” or “ISO 9126 Quality Factors”
- Sommerville: Sections “Analysis of Measurements”, “Interface Evaluation”