Software Metrics

... There is no engineering without measurements ...

Contents

- Provide an overview of state of the art;
- Discuss:
  - problems
  - solutions

Key issues

- Why measure?
- What measure?
- When measure?
- How measure?
  - Who should measure

Why Measure

- Software metrics are fundamental to:
  - Plan
  - Predict
  - Monitor
  - Control
  - Evaluate
  - products and processes
Goal

- Reduce costs.
- Improve quality.
- Control/Monitor schedule.

State of the Art ...

Measure What ... ?

- Identify:
  - Entity
  - entity attributes
- Measure:
  - height (attribute) of animals (entity)
  - size (attribute) of C++ programs (entity)

Do NOT Measure

- Attribute:
  - Height
  - Temperature
- Entity:
  - Programs
  - Processes
  - People
  - Animals

We measure attributes of entities
Software Entities

- **Process**: any activity related to software development.
- **Product**: any artifact produced during software development.
- **Resource**: money, people, hardware, software required.

Entities Attributes

- **Internal attribute** of an entity: they can be measured only based on the entity
  - *they can be measured directly*
- **External attributes** of an entity: they can be measured only wrt how the entity relates to its environment
  - *they cannot be measured directly*

### Product Entities and Attributes

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement</td>
<td>Size, Reuse, Functionality, Stability, Comprehensibility</td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td>Coupling, cohesion, modularity, size, Maintainability, quality, comprehensibility</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Size, reuse, complexity, modularity, Reliability, usability, maintainability</td>
<td></td>
</tr>
<tr>
<td>Test set</td>
<td>Size, coverage level, Quality</td>
<td></td>
</tr>
</tbody>
</table>

### Process Entities and Attributes

- **Requirements**
  - time, efforts
  - cost effectiveness
- **Design**
  - time, efforts, number of changes
  - cost effectiveness
**Resources ...**

- Personnel
  - age, cost
  - experience, productivity

- Hardware
  - memory, speed, cost
  - reliability, usability

**Empirical Relation Systems**

An attribute is characterized by a set of empirical relations $R$ among the entities $C$

$C = \{ \text{Lion}, \text{Cat}, \text{Elephant} \}$  
$R = \{ R_1, R_2, R_3 \ldots \}$

The pair $(C, R)$ is said to be an empirical relation system

**Relations**

$R_1 : (\text{Entity1, Entity2}) : \text{Entity1 IS TALLER THAN Entity2}$

$R_2 : (\text{Entity1}) : \text{Entity1 IS TALL}$

$R_3 : (\text{Entity1, Entity2, Entity3}) : \text{Entity1 IS TALLER THAN Entity2 IF STANDING ON Entity3's BACK}$

**Empirical relation System**

- Elephant IS TALLER THAN donkey
- donkey IS TALLER THAN lion
- Elephant IS TALL
- Cat IS NOT TALL
- The donkey IS TALLER THAN elephant IF STANDING ON lion BACK
Measure

A measure $M$ for an attribute is a mapping from an empirical relation system $(C,R)$ to a numerical relation system $(N,P)$ where:

- $N$ is a set of numbers
- $P$ is a set of relationships among $N$ elements

P Mapping

Each entity of $C$ is mapped into a number:
- $M(\text{cat}) = 0.5$
- $M(\text{lion}) = 1$
- $M(\text{donkey}) = 1.5$
- $M(\text{elephant}) = 2$

Relations Mapping

Each relation of $R$ is mapped into a relation of $N$

- $M(\text{Relation1})=M(\text{Entity1})">"M(\text{donkey})$
- $M(\text{Relation2})">"$
- $M(\text{Relation3})=M(\text{Entity1}*\text{Entity3})">"M(\text{Entity2})$

- "*" is the operation by which entity1 stands on entity3 back.

Consistency Condition

All empirical relations must be preserved in the numerical relation system (Representation Condition)

- $M(\text{Elephant}) > M(\text{Donkey})$
- $M(\text{Donkey}) > M(\text{Lion})$
- $M(\text{Donkey}*\text{Lion}) > M(\text{Elephant})$
Measurement Scales - Nominal

- Labeling/Classification
  - One to one mapping
    - The empirical relation system consists only of different classes; there is no notion of ordering among classes.

Nominal Example

Suppose to measure faults in a software system and to capture the location being either in requirements, design, code. We classify a fault as specification, design or code according to where the fault was firstly located.

\[
M(x) = \begin{cases} 
1 & \text{if } x \text{ is a specification fault} \\
2 & \text{if } x \text{ is a design fault} \\
3 & \text{if } x \text{ is a source code fault}
\end{cases}
\]

Measurement Scales - Ordinal

- Preference, ranking, hardness
  - Monotone increasing transformations
    - The empirical relation system consists of classes that are ordered with respect of an attribute
    - The numbers represent ranking, hence addition, subtraction and other arithmetic operations have no meaning.

Ordinal Example

We will to capture “complexity” of software modules, module complexity is classified as: “trivial”, “simple”, “moderate”, “complex” and “incomprehensible”. There is an implicit order relation: “less than”; “trivial” are simpler than “simple”, that are simpler than “moderate”, that are simpler than “complex”...

Fault severity classification (Riley 1995)

\[
M(x) = \begin{cases} 
1 & \text{if } x \text{ is trivial} \\
2 & \text{if } x \text{ is simple} \\
3 & \text{if } x \text{ is moderate} \\
4 & \text{if } x \text{ is complex} \\
5 & \text{if } x \text{ is incomprehensible}
\end{cases}
\]

\[
M(x) = \begin{cases} 
1 & \text{if } x \text{ is minor} \\
2 & \text{if } x \text{ is significant} \\
3 & \text{if } x \text{ is critical} \\
4 & \text{if } x \text{ is catastrophic}
\end{cases}
\]
Measurement Scales - Interval

- Calendar time, Centigrade/Fahrenheit temperature
  - $M' = a M + b \ (a>0)$
  - Preserve difference, not ratio
  - Addition, subtraction are acceptable but not multiplication and division.

Interval Example

We can measure temperature on Celsius or Fahrenheit scale. In other words increasing from 20 to 21 degrees in Rome in summer increase the heat in the same way that moving from 30 to 31 degrees does in Washington.

$$M = a M' + b$$

This is called an affine transformation.

Measurement Scales - Ratio

- Length, weight, time intervals, absolute temperature
  - There is a 0 element representing the total lack of the attribute.
  - All arithmetic operation are meaningful.
    - $M' = a M \ (a \ - \ positive \ scalar)$
    - Is called a ratio transformation.

Ratio Example

The length of a physical object as well as the length of software code are ratio example. we can measure in meters, feet, centimeters, inches.

To convert from one unit to the other a multiplication is required.
**Measurement Scales - Absolute**

- **Counting entities**
  - Count the elements in a set
  - There is only one possible mapping
  - All arithmetic operations are allowed

**Absolute Example**

The number of failure during testing phase.

Notice that the program length in LOC is NOT an absolute scale: we can measure it as KLOC, number of character, number of words ... LOC is a ratio measure of length of program source code attribute.

LOC is an absolute measure of the attribute "number of line of code".

"Number of year" is a ratio measure of a person's age: it cannot be an absolute measure of age, because we can also measure age in months, hours, minutes, or second.

**Statistics and Nominal Scale**

- Nominal scale defines equivalence relations
- Statistics:
  - mode, frequency, contingency coefficient
- Appropriate statistical test: non parametric

**Nominal Scale Example**

We measured faults in a software system and captured the location:

\[
M(x) = \begin{cases} 
1 & \text{if } x \text{ is a specification fault} \\
2 & \text{if } x \text{ is a design fault} \\
3 & \text{if } x \text{ is a source code fault} \\
4 & \text{if } x \text{ is a code fault} \\
5 & \text{if } x \text{ is a task fault} 
\end{cases}
\]

Mode: the most common occurring value is 3
Frequency: 3 has 3/5
Statistics and Ordinal Scale

- Ordinal scale defines equivalence and greater than relations
  - Statistics:
    - + median, percentile, Kendal W, Kendal τ

- Appropriate statistical test:
  nonparametric

Statistics and Interval Scale

- Interval scale defines equivalence, greater than relations, ratio of any intervals
  - Statistics:
    - + mean, standard deviation.

- Appropriate statistical test:
  parametric and nonparametric

Statistics and Ratio Scale

- Ratio scale defines equivalence, greater than relations, ratio of any intervals, ratio of any two scale values
  - Statistics:
    - + geometric mean.

- Appropriate statistical test:
  parametric and nonparametric

Attribute Examples

- Size
- Complexity
- Cohesion
- Coupling
- Information hiding

How to characterize these attributes?
Consider complexity ...