SOFTENG 701:
Advanced Software Engineering Development Methods
Part 2

Lecture 03a: Learning to Measure

Ewan Tempero
Department of Computer Science

Agenda

- Admin
  - Assignment?
- Part 1
  - What do metrics really measure?
  - What is measurement?
-
- Part 2
  - Measuring size
- Reading
  - Shyam R. Chidamber and Chris F. Kemerer. Towards a metrics suite for object oriented design. In *Proceedings of 6th ACM Conference on Object-Oriented Programming Systems Languages and Applications (OOPSLA)*, pages 197–211, 1991.
  - Shyam R. Chidamber and Chris F. Kemerer. A Metrics Suite for Object Oriented Design. **IEEE Transactions on Software Engineering**, 20(6):476-493, June 1994
  - See also `reading.pdf` — list of questions that might be useful for getting the most from reading a research paper

- Agenda
- Quotes
- Measurement
- Definition
- EXamples
- Representation Condition
- Typical SE Metrics
- Example: CCN
- Interpretation
- Measurement Scales
- Metrics
- Metric Problems
- Key Points

Software Engineering Measurement
by John C. Munson,
Auerbach 2003

Measurement

I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced to the stage of Science, whatever the matter may be. William Thomson, 1st Baron Kelvin (a.k.a. Lord Kelvin)

What is not measurable make measurable. Galileo Galilei (1564–1642)

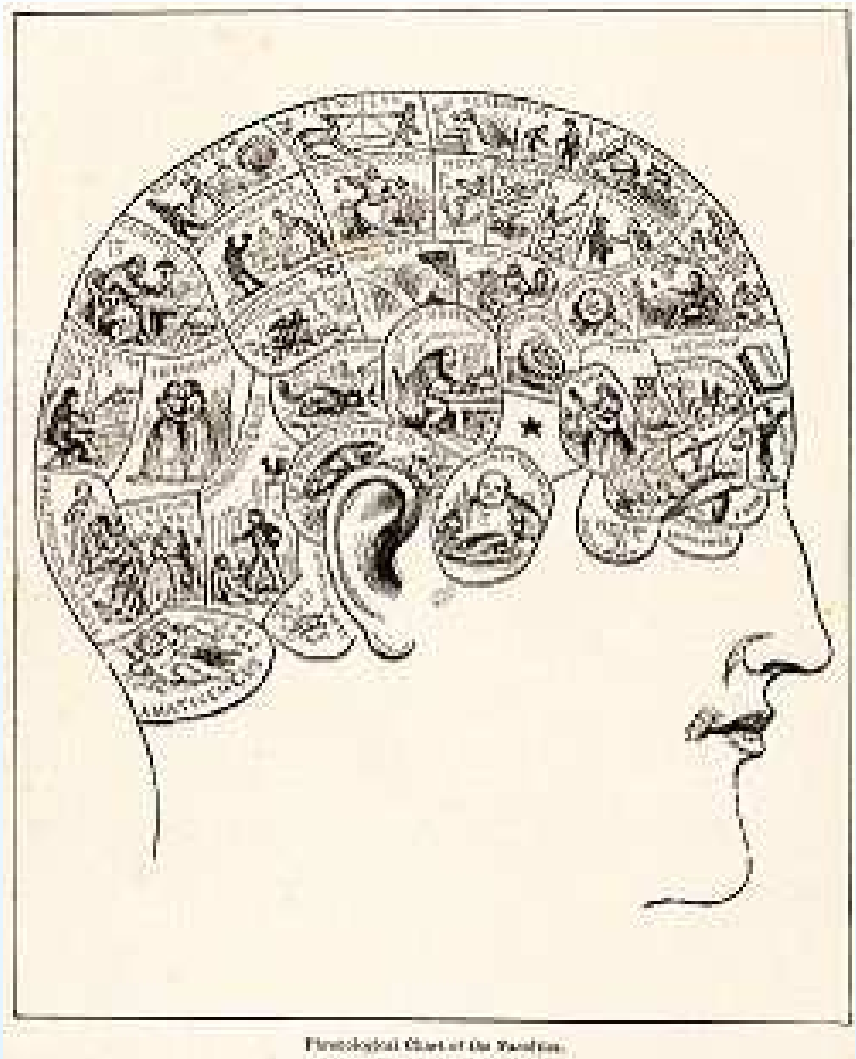
A major difference between a well developed science such as physics and some of the less well-developed sciences such as psychology or sociology is the degree to which things are measured. Fred S. Roberts

You can't control what you can't measure Tom DeMarco, (1982) Controlling Software Projects: Management, Measurement & Estimation, Yourdon Press

Measurement

- Measurement for “invisible” things such as software is not easy.
- To do so, we need a good understanding as to what measurement means.
- That understanding will help us determine whether the following “make sense”
  - “Something that produces a number”? — Measure distance between two points by walking between those two points and counting the number of fence-posts passed.
  - “Get measurements of different things and ’average’ them together” — Measure the importance of a town by adding the year of establishment, height above sea-level, and population
  - “Create a theory about how things ’must’ work, and development measurements to support it” — use the shape of bumps on someone’s skull to measure their personality

Phrenology



(From http://en.wikipedia.org/wiki/Phrenology, figure from 1883)

What is Measurement?

> Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules.

- **Metric** — description of the process to measure an attribute
- **Measurement** — can also refer to the result of the measurement process (application of the metric)

Classes of entities how tasks are done, things, what is used

Types of attributes

Internal: those that can be measured by examining the entity on its own, separate from its behaviour

External: those whose measurement must take into account both its behaviour and the environment it exists in

Example: Attribute — Height

- Entities with the height attribute: humans, horses, buildings, mountains
- Height is an *internal* attribute
- Reasonable measurements might be reported in: centimetres, hands, stories, or metres

Example: Attribute — Colour of eyes

- Entities — things with eyes
- internal attribute
- Reasonable measurements might be reported as: blue, blue-gray, blue-green, green, gray, black, . . .

Example: Attribute — Weight

- Entities with the weight attribute: humans, horses, buildings, mountains
- Weight is an _____ attribute
- Reasonable measurements might be reported in:

Example: Attribute Health

- what kind of attribute?
- measure blood pressure, weight, cholesterol — together they give some indication of health
- there are attributes that are difficult to measure “directly”
- so we measure lots of other attributes and combine them in a way that (we hope) will tell us what we want to know about the attribute we care about
- Health is a construct
- many software related attributes we care about are like this

  E.g., return on investment of reuse, depends on how “reusable” code is, which is affected by how much “polymorphic behaviour” there is, which is hard to measure (or even say what measurement of it means)

Representation Condition

- When is something a “measurement”?
  - when the value reported is the one that is “supposed to be” reported
  - when the *definition of what value is “supposed to be” reported “makes sense”*
- **empirical relationship** — the actual relationship in the real world between the entities based on some attribute

  “Mary is taller than Tom”, “Tom is heavier than Sally”, “Sally is greener than Alfred”
- **representation condition** — the relationship between the measurements given by the metric’s mapping function is the same as the empirical relationship

  “Mary is taller than Tom” $\Rightarrow$ height(Mary) $>$ height(Tom)

  height(Mary) $>$ height(Tom) $\Rightarrow$ “Mary is taller than Tom”

Constructs and Representation Condition

- Recall
  - A construct is a concept that is *quantifiable in principle* but *not directly measurable*
  - Instead, constructs are inferred from reflective indicators, direct measurements that generally correlate with our understanding of the construct
- Reflective indicators for a construct do not *measure* the construct, so cannot strictly meet the representation condition
- Nevertheless they must meet the representation condition most of the time to be true indicators $\Rightarrow$ correlation
- Demonstrating that a potential indicator correlates with the construct must be done carefully

Representation Condition: Design

- Suppose someone proposes a metric `design_quality`, how to test whether it meets the representation condition?

  Wrong way

  - Get a bunch of designs
  - Measure them according to a metric that is "known" to measure quality, e.g. larger LOC means poorer design
  - Confirm that the whenever design A is worse than design B according to LOC, that the value for `design_quality` for A is larger than that for B

A(LOC)	B(LOC)	A(d_q)	B(d_q)
1000	900	23	27
3900	800	19	59
710	800	19	17
12090	39700	258	127

  This only works if LOC really is a measure of quality, in which case the `design_qulaity` metric sn't needed!

Representation Condition: Design

- Suppose someone proposes a metric `design_quality`, where the larger the value the better the design, how to test whether it meets the representation condition?

  Right (or at least better) way

  - Get a bunch of designs
  - Compare them empirically in pairs (get experts to come to a consensus as to which design in each pair is the better design)
  - Compare the values provided by `design_quality` and confirm that they have the same relationship.

A vs B	A(d_q)	B(d_q)
better	41	27
worse	18	50
worse	14	17
better	258	127

Typical SE Metrics

- Size
  - lines of code
  - Halsted's Software Science
  - Function Points
  - WMC, RFC
- McCabe's Cyclic Complexity Number (CCN)
- Asymptotic complexity
- Test effectiveness ratio
- Coupling and Cohesion (e.g. CBO, LCOM)
- Structure (e.g. for inheritance, DIT, NOC)
- Cost Models
  - COCOMO

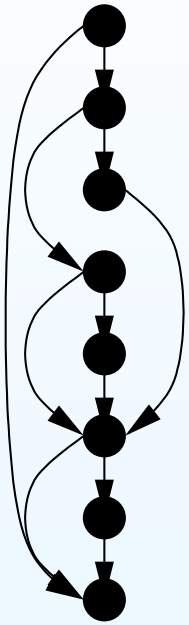
Control Flow Graphs

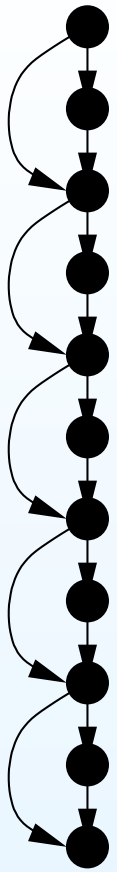
- Directed graphs can be used to model control flow of a program $\Rightarrow$ Control Flow Graph (CFG)
- *vertex* = statement, *edge* = (A, B) if control flows from statement A to B
- properties of CFGs may provide information about properties of the code
  - one property of interest: *how many ways can control flow through the code*

McCabe's Cyclomatic Complexity Number (CCN)

- proposed as a "measurement" for code complexity
- measures the number of *linearly independent* paths through the control flow graph
- $v(F) = e - n + 2$, F the CFG of the code, n the number of vertices, e the number of edges
- Intuition — the larger the CCN the "more complex" the code
- Various sources recommend a CCN of no more than 10-15

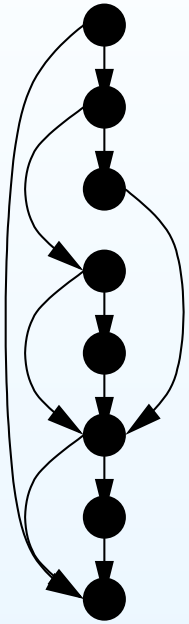
(counter) Example 1



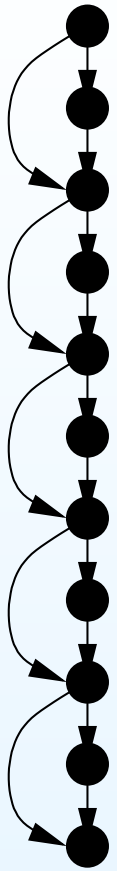


(counter) Example 1

```
if (a1) {
 if (a2) {
  ...
 } else if (a3) {
  ...
 }
 if (a3) {
  ...
 }
}
```

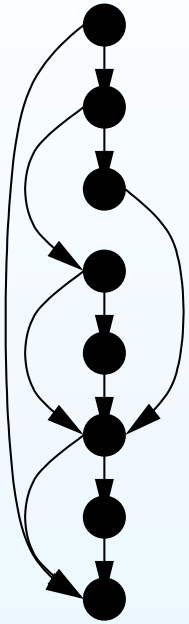


```
 if (b1) {
  ...
}
 if (b2) {
  ...
 }
 if (b3) {
  ...
 }
 if (b4) {
  ...
 }
 if (b5) {
  ...
 }
```

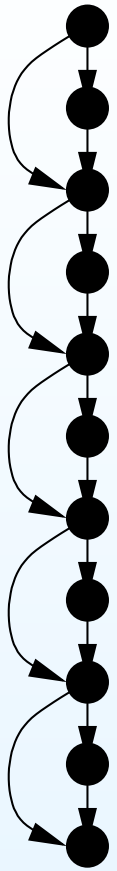


(counter) Example 1

```
if (a1) {
 if (a2) {
  ...
 } else if (a3) {
  ...
 }
 if (a3) {
  ...
 }
}
```



```
 if (b1) {
  ...
}
 if (b2) {
  ...
 }
 if (b3) {
  ...
 }
 if (b4) {
  ...
 }
 if (b5) {
  ...
 }
```

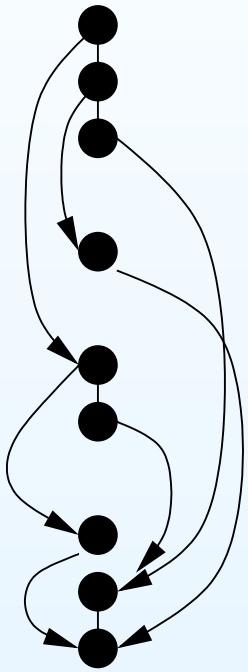


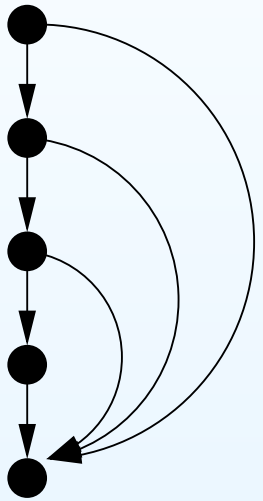
CNN$= 11 - 8 + 2 = 5$

CNN$= 15 - 11 + 2 = 6$

(counter) Example 2

```
if (a)
   if (b)
      goto d:
   else
      goto e:
else
   if (c)
      goto d:
   else
      goto e:
d: print "Hi. "
e: print "How are you?"
```

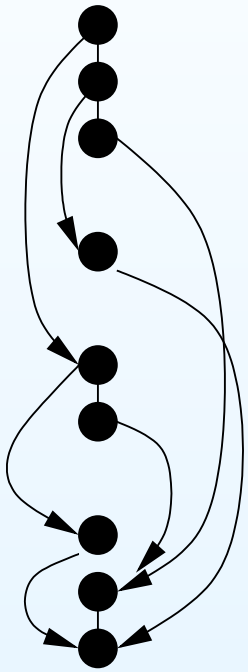




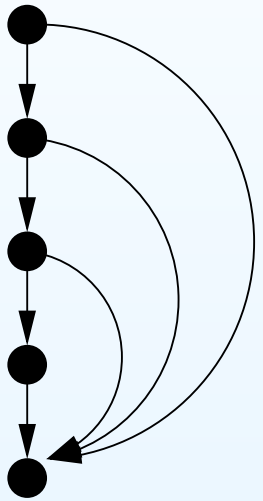
```
if (u)

   if (v)

      if (w)

         print "Hi"

print "How are you?"
```

(counter) Example 2

```
if (a)
   if (b)
      goto d:
   else
      goto e:
else
   if (c)
      goto d:
   else
      goto e:
d: print "Hi. "
e: print "How are you?"
```



CNN$= 11 - 9 + 2 = 4$



```
if (u)
   if (v)
      if (w)
         print "Hi"
print "How are you?"
```

CNN$= 7 - 5 + 2 = 4$

(counter) Example 3

- From graph theory, $v(f) = d + 1$, where d is the number of predicate nodes (tests/conditions)

```
public static boolean isPrimeA(int n) {
  for (int i = 2; i < n; i++) {
    if (n % i == 0) {
      return false;
    }
  }
  return true;
}

public static boolean dividableBy4(int n) {
  if (n % 2 == 0) {
    if ((n/2) % 2 == 0) {
      return true;
    }
  }
  return false;
}
```

How “good” is CCN as a metric?

- As a means to measure the number of linearly independent paths through a method $\Rightarrow$ very good
- As a means to measure “complexity”, not so much.
  - “complexity” is a construct, so cannot measure
  - some evidence to suggest it is a reasonable indicator, but so it “size”

The “meaning” of measurements

The patient is 1 metre tall.

- What can we conclude from this statement?
- Do we have enough information to conclude anything useful?

Entity Population Models

- These models describe the distribution of the data for a given metric — identify typical values
- Examples:
  - Core body temperature for humans is 37 degrees
  - The average male height in the age range 19-45 years in New Zealand is 177cm (WIKIPEDIA)
  - Girls grow on average 6.5cm/year between the ages of 2-13 (WIKIPEDIA)
- We use our understanding of typical values to make decisions

$\Rightarrow$ if a "typical" measurement from a metric does not tell us anything useful, it's not a useful metric

Measurement Scales

- characterises what can be done with the measurements, in particular, what kinds of statistical analysis is appropriate

  Nominal measurements are categories with no ordering or other structure

  Ordinal measurements are categories with a defined order (rank), but don't know "distance" between the categories

  Interval measurements are an equal distance apart

  Ratio measurements have a "true zero" (absolute smallest value)

  Absolute measurements are a count

Scales: Examples

Nominal colour of eyes, (yes, no), (male, female), bus route numbers

Ordinal (very cold, cold, cool, warm, hot, very hot), (too cold, just right, too hot), house numbers (in New Zealand at least)

Interval Year, Fahrenheit and celsius temperature scales

Ratio kelvin temperature scale, distance/length, weight, speed

Absolute counts

Scales: statistics

Nominal cannot perform any arithmetic operations (addition, multiplication, etc), but can compute mode, frequency distributions

Ordinal cannot perform any arithmetic operations (addition, multiplication, etc), but can compute median (and mode), and such things as “non-parametric analysis of variance”

Interval can perform some forms of arithmetic, but not ratios, so average, standard deviation, regression, and other parametric techniques

Ratio can perform ratios, and any other arithmetic technique

Absolute specialised ratio scale

Example

The lecturer presented the course content in an interesting manner.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	Not Applicable
○	○	○	○	○	○

- Does someone think this is a form of measurement?
- Is this really a form of measurement?
- What entities are being measured?
- What is the attribute?
- What kind of attribute is it?
- What *measurement* should be reported?
- How can the measurements be used?
- What do the values (e.g. “Neutral”) mean?

Metric Definitions

- the specification of what the measurement is
- includes mapping relationship between attributes of entities being measured to "numbers or symbols" (the measurements)
- numbers or symbols often referred to as units of measurement
- units of measurement have a scale
- metric should meet the representation condition
- entity population model to interpret measurements

Other considerations

- there needs to be a way to get the measurements — the *instrument*
- how the instrument is used needs to be specified — the *measurement protocol*

Typical problems with software metrics

- Metric is not fully defined
- Metric does not meet fundamental rules for measurement (e.g. what is the unit)
- Measurements are not used correctly according to their scale
- Metric is not an indicator of the construct
  - does not have face validity

Key Points

- Metrics describe how to associate measurements to attributes of entities
- Most software metrics have been proposed to measure some quality attribute (but can at best be no more than reflective indicators)
- Creating a metric for software requires more than just coming up with a way to associate numbers with code
  - E.g. need entity population models