CSE 4621 / SWE 5620
Software Metrics

The Basics of Measurement
Chapter -2-

Khaled Slhoub, PhD

Difficult Software Measurement

Florida Tech

1. How much must we know about an attribute before it is reasonable to consider measuring it? For instance, do we know enough about "complexity" of programs to be able to measure it?
2. How do we know if we have really measured the attribute we wanted to measure? For instance, does a count of the number of "bugs" found in a system during integration testing measure the quality of the system? If not, what does the count tell us?
3. Using measurement, what meaningful statements can we make about an attribute and the entities that possess it? For instance, is it meaningful to talk about doubling a design's quality? If not, how do we compare two different designs?
4. What meaningful operations can we perform on measures? For instance, is it sensible to compute average productivity for a group of developers, or the average quality of a set of modules?

Empirical Relations

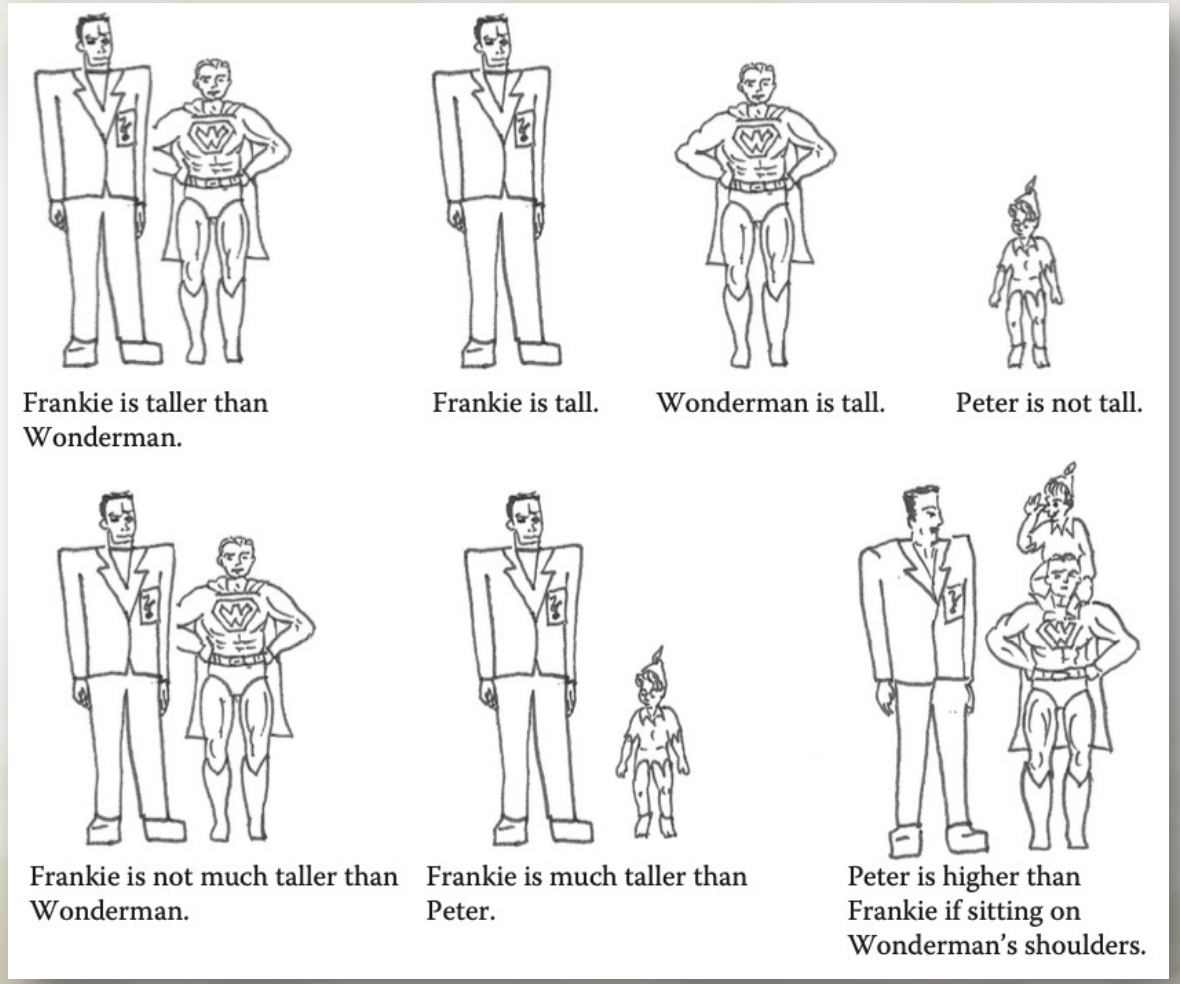
In any measurement activity, there are rules to be followed

the data we obtain as measures should represent attributes of the entities we observe

manipulation of the data should preserve relationships that we observe among the entities

Consider the way we perceive the real world. We tend to understand things by **comparing** them, not by assigning numbers to them.

The Representational Theory of Measurement



Some empirical relations for the attribute *height*

Empirical Relations

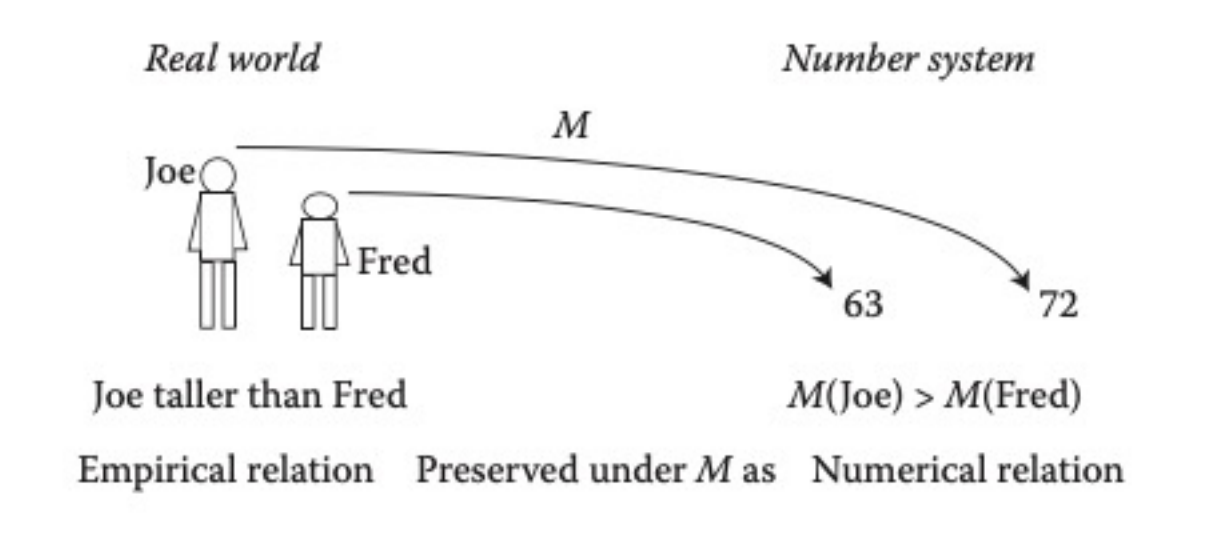
An empirical relation is one for which there is a reasonable consensus about which pairs are in the relation

X and Y are empirical objects

- They may have an empirical relation, such as "X is more difficult to maintain than Y"
- X ●> Y
  - is analogous to X is greater than Y, but we mean that a property of X is more than a property of Y, e.g. X is more difficult to maintain than Y, X is buggier than Y
- X ≈ Y
  - A property of X is the same as (equivalent to) a property of Y, for example, X is equally difficult to maintain
- X ●≥ Y
  - A property of X is more than or equal to the property of Y

Measurement Mapping

- We can think of these relations as mappings from the empirical, real world to a formal mathematical world.
  - Empirical relation preserved under measurement M as numerical relation



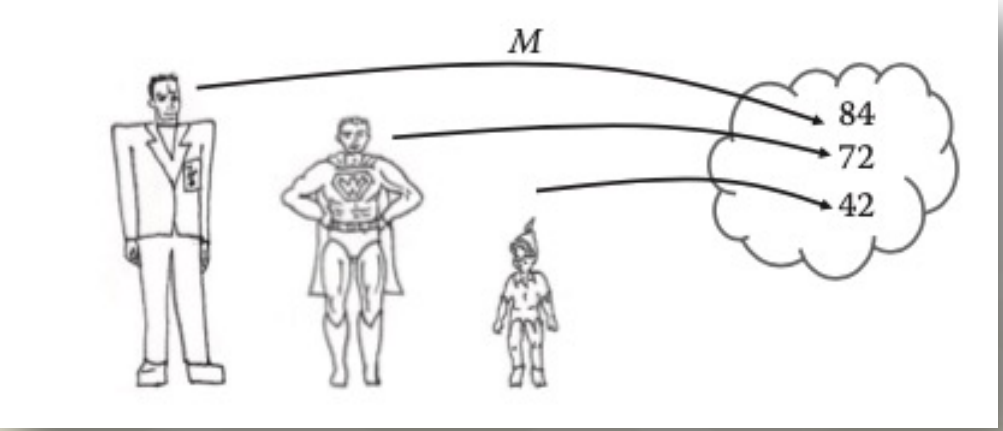
Whenever Joe is taller than Fred, then *M*(Joe) must be a bigger number than *M*(Fred)

Measurement Mapping

- We can think of these relations as mappings from the empirical, real world to a formal mathematical world.

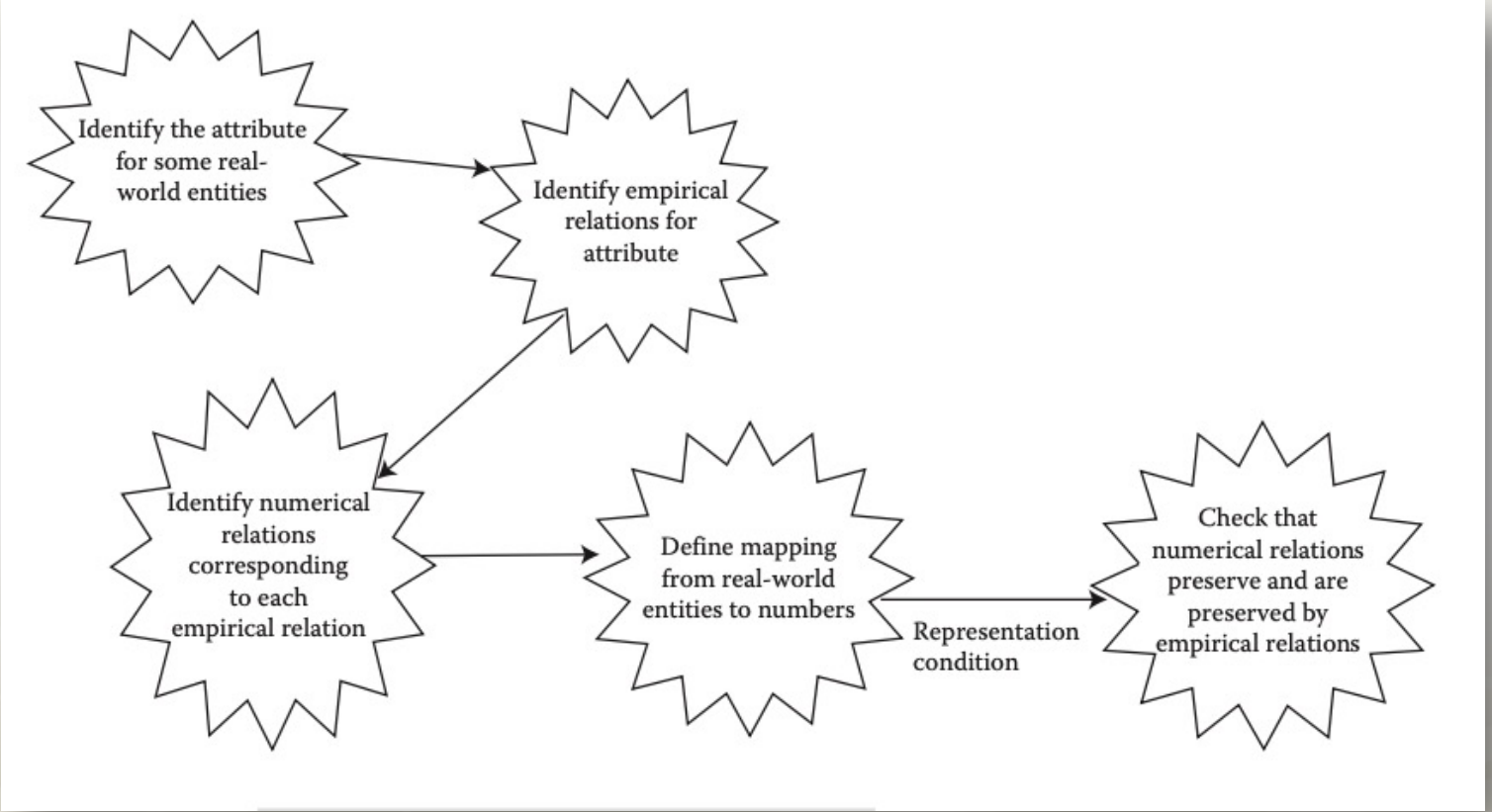
  Wonderman is mapped to the real number 72 (i.e., M(Wonderman) = 72), Frankie to 84 (M(Frankie) = 84), and Peter to 42 (M(Peter) = 42).

- Frankie is taller than Wonderman, *M*(Frankie) > *M*(Wonderman).
- Wonderman is tall, *M*(Wonderman) = 72 > 70.
- Frankie is much taller than Peter, *M*(Frankie)=84>57=*M*(Peter)+ 15. Similarly, Wonderman is much taller than Peter, *M*(Wonderman) = 72 > 57 = *M*(Peter) + 15.
- Peter is higher than Frankie when sitting on Wonderman's shoulders,
- and 0.7*M*(Peter) + 0.8*M*(Wonderman) = 87 > 84 = *M*(Frankie)



we can define the mapping as a *measure* for the attribute.

Key Stages of Measurement Process



Definition of Measurement

- "**Formally**, we define measurement as a mapping from the empirical world to the formal, relational world. Consequently, a measure is the number or symbol assigned to an entity by this mapping in order to characterize an attribute."

- **Chapter 1** "Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way so as to describe them according to clearly defined rules."

Rules of the Mapping

- The real world is the **domain** of the mapping
- The mathematical world is the **range** of the mapping
- When we map the attribute to a mathematical system, we have many choices for the mapping and the range. We can use real numbers, integers, or even a set of non-numeric symbols.
- There must be a rule that clearly specifies the mapping, e.g., Felix is 11 (years old implied) or LOC (clear rules that define loc).

Subjective Assessment

- Often, the empirical relations for an attribute are not agreed upon, especially when they reflect personal opinion or preference , e.g., soda, ice cream, drink.
- We can still perform a subjective assessment, but the result may not be a measure.
- **This consistency of measurement is very important**

Likert Scale

Give the respondent a statement with which to agree or disagree. Example:

This software program is reliable.

Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly Disagree

Forced Ranking

Give n alternatives, ordered from 1 (best) to n (worst). Example:

Rank the following five software modules in order of maintenance difficulty, with 1 = least complex, 5 = most complex:

—	Module A
—	Module B
—	Module C
—	Module D
—	Module E

Verbal Frequency Scale

Example: How often does this program fail?

Always	Often	Sometimes	Seldom	Never

We strive to keep our measurements objective. By doing so, we make sure that different people produce the same measurement results,

So, Software Metrics Will Involve:

- Empirical relations among empirical properties (attributes)
- Functions that map empirical properties to real numbers in ways that "preserve" the empirical relations

There is nothing wrong with using several representations for the same attribute

OR

using the same representation in different ways,

	Entity	Attribute	Measure
1	Completed project	Duration	Months from start to finish
2	Completed project	Duration	Days from start to finish
3	Program code	Length	Number of lines of code (LOC)
4	Program code	Length	Number of executable statements
5	Integration testing process	Duration	Hours from start to finish
6	Integration testing process	Rate at which faults are found	Number of faults found per KLOC (thousand LOC)
7	Test set	Efficiency	Number of faults found per number of test cases
8	Test set	Effectiveness	Number of faults found per KLOC (thousand LOC)
9	Program code	Reliability	Mean time to failure (MTTF) in CPU hours
10	Program code	Reliability	Rate of occurrence of failures (ROCOF) in CPU hours

Measurement and Models

- A model is an abstraction of reality, allowing us to strip away detail and view an entity or concept from a particular perspective
- Cost models permit us to examine only those project aspects that contribute to the project's final cost
- Models come in many different forms: as equations, mappings, or diagrams
- The model of the mapping should also be supplemented with a model of the mapping's domain—that is, with a model of how the entity relates to its attributes.
- Example: LOC (comment lines, many statements on the same line, data declarations, etc.)

Direct SW Measures

- Direct mappings from attribute to number, and we use the number to answer questions or assess situations
- Involves no other attribute or entity
- The following direct measures are commonly used in software engineering:
  - Size of source code (measured by LOC)
  - Schedule of the testing process (measured by elapsed time in hours)
  - Number of defects discovered (measured by counting defects)
  - Time a programmer spends on a project (measured by months worked)

Indirct/Derived SW Eng Measures

- Complex relationships among attributes, or when an attribute must be measured by combining several of its aspects
- A model of how to combine the related measures is used
- The following derived measures are commonly used in software engineering:

Programmer productivity	LOC produced/person-months of effort
Module defect density	Number of defects/module size
Defect detection efficiency	Number of defects detected/total number of defects
Requirements stability	Number of initial requirements/total number of requirements
Test coverage	Number of test requirements covered/total number of test requirements
System spoilage	Effort spent fixing faults/total project effort

Direct Measurement Example

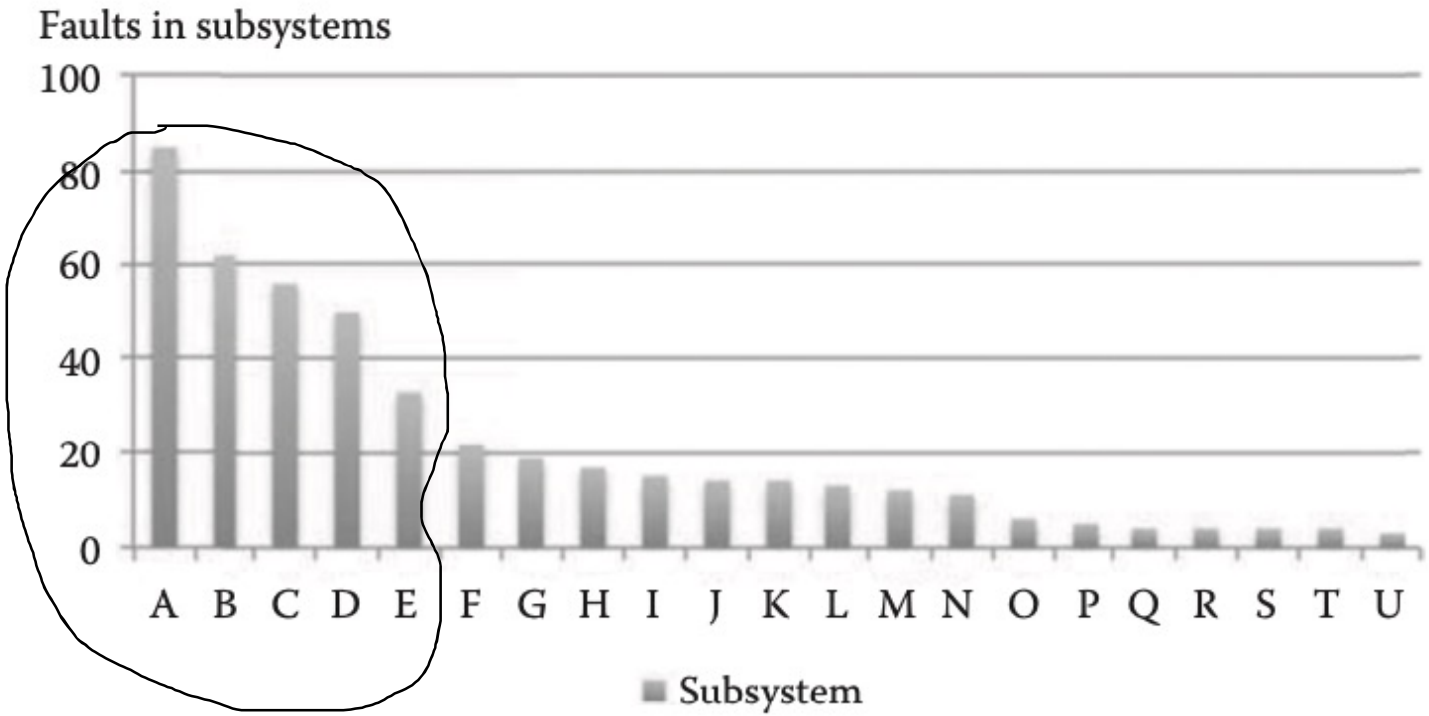


FIGURE 2.10 Using direct measurement to assess a product (from a major system made up of several subsystems).

measuring number of faults (direct measurement) leads to identification of 5 problem areas.

But then …

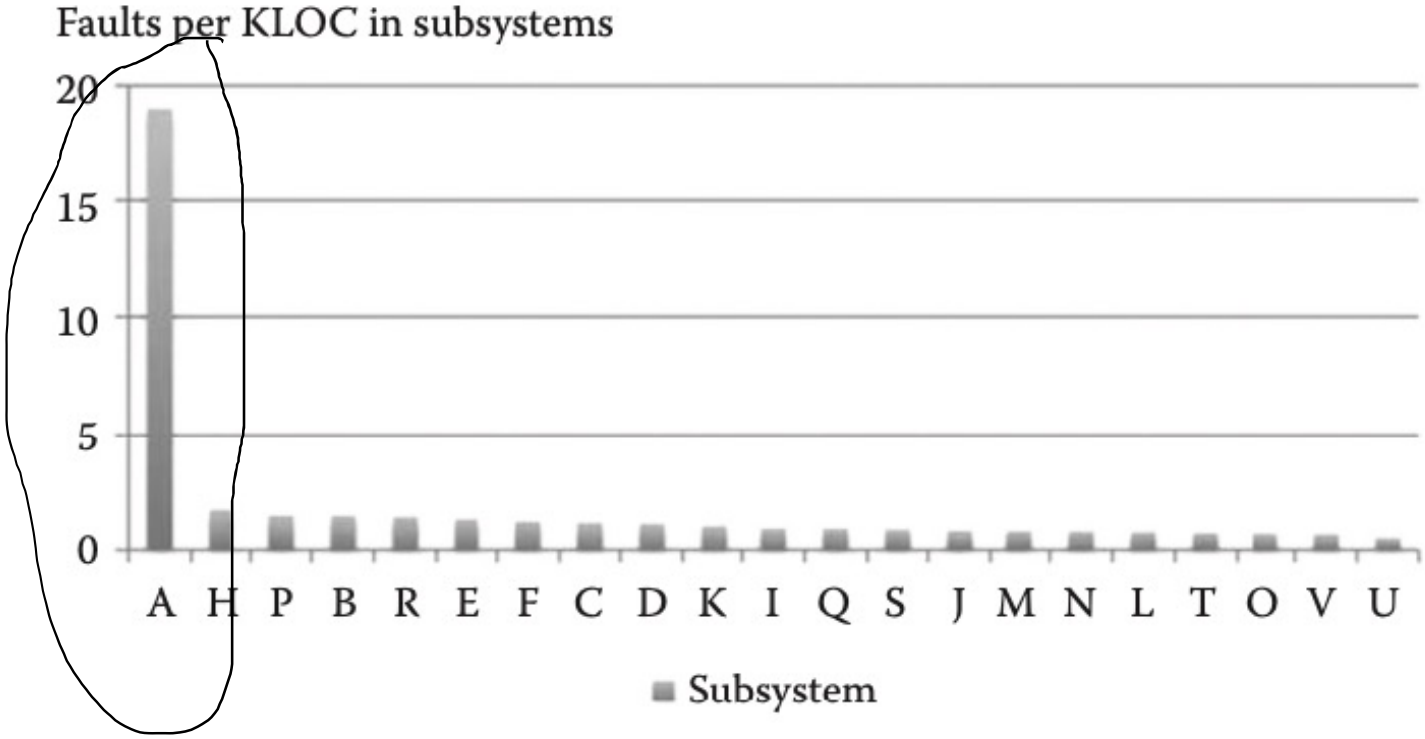


FIGURE 2.11 Using derived measurement to assess a product.

However, measuring faults per KLOC (indirect measurement) leads to identification of only one problem area.

Derived measurement is often useful in making visible the interactions between direct measurements.

Measurement Scales

- The purpose of performing the mapping is to be able to manipulate data in the numerical system and use the results to draw conclusions about the attribute in the empirical system
- The differences among the mappings can restrict the kind of analysis we can do
- To understand these differences, we introduce the notion of a **measurement scale**
- **We use the scale to help us understand which analyses are appropriate.**

Measurement Scales

- A measurement scale is a set of predefined symbols or values in order to represent certain common measures.

- A scale is an abstract measurement tool for measuring defined common attributes of entities.

Measurement Scales

- Components of a measurement system:

  m = <attribute, scale, unit>

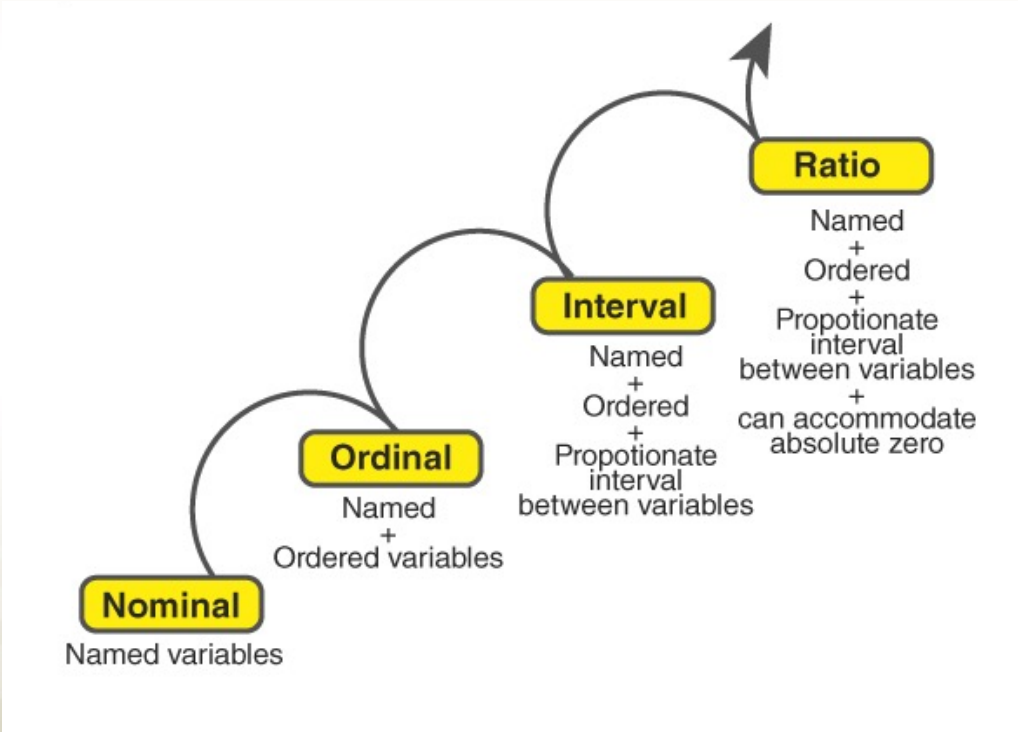
  - ***Attribute*** is what is being measured (e.g., size of a program)
  - ***Scale*** is the standard and scope of measurement (e.g., nominal, ordinal, ratio scale, etc.)
  - ***Unit*** is the physical meaning of scale (e.g., a positive integer, a symbol, etc.)
- Determining the value of an attribute of an entity
- Determining the class of entities to which the measurement relates

Levels of Measurement

Different representations for a given empirical relation system

to determine which representation is the most suitable for measuring an attribute of interest?

Quantitative or Qualitative: Result of measurement is either quantitative (represented by number values) or qualitative (represented by qualitative values or range intervals)



Nominal Scale

- Define classes/categories, then place each entity in a particular class/category, based on the value of the attribute.
- **Characteristics:**
  - The empirical relation system consists only of different classes; there is **no notion of ordering** among the classes
  - Any distinct numbering or symbolic representation of the classes is an acceptable measure, but there is **no notion of magnitude** associated with the numbers or symbols
- Nominal-scale measurement places elements in a classification scheme. The classes are not ordered; even if the classes are numbered from 1 to n for identification, there is no implied ordering of the classes.
- **Example:**
  - Classification of cars based on their color
  - Requirement types (interface, database, ...),
  - Classification of SW faults based on location (spec, design, code)

Common Mistake with Nominal Scale

Florida Tech

- Assigning numbers and then adding, subtracting, multiplying, dividing:

Language	Symbol	Number of Projects
C++	1	3
Objective C	2	10
COBOL	3	8
Assembler	4	5

Average language: (1*3 + 2*10 + 3*8 + 4*5)/26 = 2.6

Common Mistake with Nominal Scale

- Comparing when ranking does not exist

Error type	Designation
Interface error	1
Timing error	2
User error	3

- Project manager: "Last month, our average error was a 2.6, and this month it's a 1.5, so we are getting better!"

Ordinal Scale

- Useful to augment the nominal scale with information about an ordering of the classes or categories
- The ordering leads to analysis not possible with nominal measures
- **Characteristics:**
  - The empirical relation system consists of classes that are ordered with respect to the attribute.
  - Any mappings that preserve a ranking or ordering is acceptable -- sometimes called ordered categories
  - The numbers represent ranking only, so addition, subtraction, and other arithmetic operations have no meaning–no concept of distance between pairs of objects
- **Example:**
  - Language ability (Beginner, Intermediate, Fluent)
  - Quality of requirements (Low, Moderate, High)
  - Measuring software complexity (Trivial, Simple, Moderate, Complex)

Common Mistake with Ordinal Scale

- Averaging (as with nominal):
  - "Our organization has an average SEI maturity level of 1.8, so we are almost at level 2....."
    - There is no concept of distance: how far is it from level 1 to level 2?

Interval Scale

- Interval scale carries more information than ordinal and nominal scale. It captures information about the size of the intervals that separate the classes, so that we can in some sense understand the size of the jump from one class to another.
- **Properties:**
  - An interval scale **preserves order**, as with an ordinal scale.
  - An interval scale **preserves differences but not ratios**. That is, we know the difference between any two of the ordered classes in the range of the mapping, but computing the ratio of two classes in the range does not make sense.
  - Addition and subtraction are acceptable on the interval scale, but not multiplication and division.
- **Example:**
  - Temperature
  - Project scheduling (Spec 3w, Design 4w, Coding 4w, Testing 3w)

Example of Interval Scale

Recall the five categories of complexity described in Example 2.16. Suppose that the difference in complexity between a trivial and simple system is the same as that between a simple and moderate system. Then any interval measure of complexity must preserve these differences. Where this equal step applies to each class, we have an attribute measurable on an interval scale. The following measures have this property and satisfy the representation condition:

$$M_1(x) = \begin{cases} 1 \text{ if } x \text{ is trivial} \\ 2 \text{ if } x \text{ is simple} \\ 3 \text{ if } x \text{ is moderate} \\ 4 \text{ if } x \text{ is complex} \\ 5 \text{ if } x \text{ is incomprehensible} \end{cases} \qquad M_2(x) = \begin{cases} 0 \text{ if } x \text{ is trivial} \\ 2 \text{ if } x \text{ is simple} \\ 4 \text{ if } x \text{ is moderate} \\ 6 \text{ if } x \text{ is complex} \\ 8 \text{ if } x \text{ is incomprehensible} \end{cases}$$

$$M_3(x) = \begin{cases} 3.1 \text{ if } x \text{ is trivial} \\ 5.1 \text{ if } x \text{ is simple} \\ 7.1 \text{ if } x \text{ is moderate} \\ 9.1 \text{ if } x \text{ is complex} \\ 11.1 \text{ if } x \text{ is incomprehensible} \end{cases}$$

Common Mistake with Interval Scale

- Ratios of measures

  it is usually 60° on a summer's day in London, while it may be 90° on the same day in Orlando

if you said, "it is one-third as hot in London as Orlando," you would be incorrect

Also, we cannot say it is 30% hotter in Orlando than in London.

All we can say is that the summer in Orlando is warmer than London.

Ratio Scale

- Sometimes we would like to be able to say that one liquid is twice as hot as another, or that one project took twice as long as another. This needs the ratio scale, which is the most useful scale of measurement, and quite common in the physical sciences
- **Characteristics:**
  - It is a measurement mapping that preserves ordering, preserves size of intervals between entities, and preserves ratios between entities
  - There is a zero element, representing total lack of the attribute.
  - The measurement mapping must start at zero and increase at equal intervals, known as units.
  - All arithmetic can be meaningfully applied to the classes in the range of the mapping
- **Example:**
  - Number of faults, Number of lines of code, Measuring execution time of a program, Years of experience, age, …

Absolute Scale

- The absolute scale is the most restrictive of all. For any two measures, M and M', here is only one way in which the measurement can be made, so *M* and *M′* must be equal.
- **Characteristics:**
  - The measurement for an absolute scale is made simply by counting the number of elements in the entity set.
  - **The attribute always takes the form "number of occurrences of x in the entity."**
  - There is only one possible measurement mapping.
  - All arithmetic analysis of the resulting count is meaningful.
- **Example:**
  - Number of failures observed during integration testing , but the reliability is not.
  - Number of people working on a software project is absolute, but their productivity is not.

Example of Absolute Scale

- A common mistake is to assume that LOC is an *absolute* scale measure of a program length, because it is obtained by counting
  - many way to count: number of lines, thousands line of code, number of characters, number of bytes, …
  - LOC is an absolute scale measure of the attribute “number of lines of code” of a program.

1.A common mistake is to assume that LOC is an *absolute* scale measure of length, because it is obtained by count- ing. However, it is the *attribute* (as characterized by empirica

Common Mistake with Absolute Scale

Florida Tech

- A common mistake is to assume that LOC is an absolute scale measure of length, because it is obtained by counting. However, it is the attribute (as characterized by empirical relations) that determines the scale type.
- As we have seen, the length of programs cannot be absolute, because there are many different ways to measure it (such as LOC, thousands of LOC, number of characters, and number of bytes). It is incorrect to say that LOC is an absolute scale measure of program length.
- **However, LOC is an absolute scale measure of the attribute "number of lines of code" of a program.**
- For the same reason, "number of years" is a ratio scale measure of a person's age; it cannot be an absolute scale measure of age, because we can also measure age in months, hours, minutes, or seconds.

Summary of Scale Measures

Florida Tech

Levels of Measurement

Nominal	Ordinal	Interval	Ratio
"Eye color"	"Level of satisfaction"	"Temperature"	"Height"
Named	Named	Named	Named
	Natural order	Natural order	Natural order
		Equal interval between variables	Equal interval between variables
			Has a "true zero" value, thus ratio between values can be calculated

Source: https://www.statology.org/levels-of-measurement-nominal-ordinal-interval-and-ratio/

Questions to Think About

Florida Tech

- List 5 examples of attributes of software product, process, resource, or impact that we measure on a **nominal scale**. For each one, why do we use a nominal scale rather than one of the others?
- List 5 examples of attributes of software product, process, resource, or impact that we measure on an **ordinal scale**. For each one, why do we use an ordinal scale rather than one of the others?
- List 5 examples of attributes of software product, process, resource, or impact that we measure on an **interval scale**. For each one, why do we use an interval scale rather than one of the others?
- List 5 examples of attributes of software product, process, resource, or impact that we measure on a **ratio scale**. For each one, why do we use a ratio scale rather than one of the others?

Exercise 1

Determine the best "scale" for each of the following measurements using each of the Nominal, Ordinal, Interval and Ratio scales only once.

- Measuring execution time of a program
- Classification of objects based on their color
- Measuring duration of various phases of projects (Project scheduling)
- Measuring complexity of many software modules by defining 4 complexity classes (Trivial, Simple, Moderate, Complex)
- Measuring complexity of many software modules by defining cyclomatic complexity metrics

Exercise 1 (cont.)

Determine the best "scale" for each of the following measurements using each of the Nominal, Ordinal, Interval and Ratio scales only once.

- Software products categorized according to their compatibility with the operating system (i.e. Windows, Linux, MacOS, DOS, etc.).
- Internet services categorized according to their relevant technologies (i.e. dial-up, DSL, high-speed, wireless, etc.)
- Measuring attitudes towards an Internet service (say, on an n-point rating, n = 0 to 10).
- Measuring credit scores which are ranged from 300 to 850
- Jane who is 6 feet tall is 1.5 times taller than Karen who is 4 feet tall.

Exercise 1 (cont.)

Determine the best "scale" for each of the following measurements using each of the Nominal, Ordinal, Interval and Ratio scales only once.

- Measuring attitude towards Internet services, if the evaluation scores are numerically meaningful so the difference between a rate of 3 and a rate of 6 is exactly the same as the difference between a rate of 7 and a rate of 10.
- Measuring attitude towards Internet services, if the differences between the data values are numerically meaningful and equal. In addition, a score of zero implies either the full absence of the service being evaluated or the full dissatisfaction with it.
- The number of hours per day you spent watching TV

Exercise 2

Florida Tech

Suppose that you are asked to study various software development tools and recommend the best three to your company. The following table shows a list of available development tools.

Tool Name/Vendor	Languages Supported	Platforms	Features
Bean Machine IBM	Java	Windows, OS2, Unix	Best: Visual applet and JavaBean generation
CodeWarrior Pro Metrowerks	Java, C, C++, Pascal	Unix, Windows, Mac	Best: if you need to support Unix, Windows, and Mac platforms
Java Workshop Sun Microsystems	Java	Solaris, Windows	Better: Written 100% in Java; tools based on a web browser metaphor
JBuilder Imprise	Java	Windows, AS400	Better: database support
Visual Cafe for Java Symantec	Java	Windows	Good: multithreaded debugger
VisualAge IBM	Java	Unix, Windows	Good: includes incremental compiler and automatic version control
Visual J++ Microsoft	Java	Windows	Fair: All the bells and whistles for Windows

Exercise 2 (cont.)

Florida Tech

What are the entities, attributes and their values in your model?

Entity	*Attribute*	*Value*
Development Tool	Language supported	Java, C, C++, Pascal
	Platform	Win, Unix, Mac, OS2, AS400
	Feature	Fair, Good, Better, Best

Exercise2 (cont.)

Florida Tech

What is the best scale for each of the attributes you defined?

Entity	*Attribute*	*Value*	*Scale*
Development Tool	Language supported	Java, C, C++, Pascal	Nominal
	Platform	Win, Unix, Mac, OS2, AS400	Nominal
	Feature	Fair, Good, Better, Best	Ordinal

Meaningfulness in Measurement

Florida Tech

- Understanding scale types enables us to determine when statements about measurement make sense.

  we must remember that the analysis is constrained by the scale type

- Can we deduce meaningful statements about the entities being measured?
- Which of the following statements are meaningful?
  - The number of errors discovered during the integration testing of program X was at least 100.
  - The cost of fixing each error in program X is a least 100.
  - A semantic error takes twice as long to fix as a syntactic error.
  - A semantic error is twice as complex as a syntactic error.

Meaningfulness in Measurement

- Our intuitive notion of a statement's meaningfulness involving measurement is quite distinct from the notion of the statement's truth.
  - The President of the United States is 125 years old (a meaningful statement about the age measure)
  - A statement involving measurements is meaningful, if its truth or or falsity statement remains unchanged under any **admissible transformation**
- Formal definition of **Meaningfulness** is "*a "statement involving measurement is meaningful if its truth value is invariant of transformations of allowable scales."*

Meaningfulness in Measurement

- **Fred is twice as tall as Mary**
  - measures are at least on the ratio scale, because it uses scalar multiplication as an admissible transformation
    - M(Fred) = 2*M(Mary)
    - M'(Fred) = 2*M'(Mary)
  - This consistency of truth is due to the relationship $M = aM'$ for some positive number a.

- **The temperature in Tokyo today is twice that in London**
  - The statement implies ratio scale but is not meaningful, because we measure (air) temperature only on two scales, Fahrenheit and Celsius. $F = C * 9/5 + 32$
    - Tokyo: 40 degrees C and London: 20 degrees C
    - Tokyo: 104 degrees F and London: 68 degrees F

Meaningfulness in Measurement

- **The difference in temperature between Tokyo and London today is twice what it was yesterday**
  - This statement implies that the distance between two measures is meaningful, a condition that is part of the interval scale
  - The statement is meaningful, because Fahrenheit and Celsius are related by the affine transformation $F = 9/5C + 32$
    - Yesterday: 35 degrees C in Tokyo and 25 degrees C in London: 10 diff
    - Today: 40 degrees C in Tokyo and 20 degrees C in London: 20 diff

    Transform these temperatures to the Fahrenheit scale
    - Yesterday: 95 degrees C in Tokyo and 77 degrees C in London: 18 diff
    - Today: 104 degrees C in Tokyo and 68 degrees C in London: 36 diff

Meaningfulness in Measurement

– **Failure x is twice as critical as failure y**
  - This statement is not meaningful, since we have only an ordinal scale for failure criticality

Failure Class	Mapping *M*	Mapping *M'*
$Class_1$	1	3
$Class_2$	3	4
$Class_3$	6	5
$Class_4$	7	10

  - Suppose y is in class2 and x in class3. Notice that M(x) = 6 and M(y) = 3 while M'(x) = 5 and M'(y) = 4
  - In this case, the statement is true under *M* but false under *M'*

Statistical Operations on Measures

- The scale type of a measure affects the types of operations and statistical analyses that can be sensibly applied to the data
- If we want to know something about how the whole data set is distributed.
  - Two basic measures used to capture this information: measures of central tendency and measures of dispersion.
  - e.g., nominal and ordinal measures do not permit computation of average and standard deviation
  - the notion of mean is not meaningful for nominal and ordinal measures.

Statistical Operations on Measures

We would like to know which of the two systems has the higher average understandability

Using **M**, the **mean** of the X values is 2.6, while the mean of the Y values is 3.1

using **M'**, the **mean** of the *X* values is 3.6, while the mean of the *Y* values is 3.1

Using both **M** and **M'**, the **median** of the *Y* values (in both cases 3) is greater than the median of the *X* values (in both cases 2)

Using **M''**, the **median** of the *Y* values, 69, is still greater than the median of the *X* values, 3.8.

measure of understandability

	Trivial	Simple	Moderate	Complex	Incomprehensible
M	1	2	3	4	5
M'	1	2	3	4	10

TABLE 2.7 Different Measure M''

	Trivial	Simple	Moderate	Complex	Incomprehensible
M''	0.5	3.8	69	104	500

x_1 trivial
x_2 simple
x_3 simple
x_4 moderate
x_5 incomprehensible

while *Y*'s seven modules have understandability

y_1 simple
y_2 moderate
y_3 moderate
y_4 moderate
y_5 complex
y_6 complex
y_7 complex

Summary of Scale Measures

Provides:	Nominal	Ordinal	Interval	Ratio
The "order" of values is known		✔	✔	✔
"Counts," aka "Frequency of Distribution"	✔	✔	✔	✔
Mode	✔	✔	✔	✔
Median		✔	✔	✔
Mean			✔	✔
Can quantify the difference between each value			✔	✔
Can add or subtract values			✔	✔
Can multiple and divide values				✔
Has "true zero"				✔

Source: https://www.mymarketresearchmethods.com/types-of-data-nominal-ordinal-interval-ratio/

Exercise 2

You ask 29 survey participants to indicate their level of agreement with the statement below:

The new reporting way added to the X app is easy to use.

• Strongly disagree	2
• Disagree	2
• Neither disagree nor agree	8
• Agree	12
• Strongly agree	5

Find Mode (Nominal data), **Agree** (most frequently appearing value)

Find Median (Ordinal data) **Agree** (the middle of your data set)

Find Mean (Ordered data) **N/A**

Exercise 4

Florida Tech

Determine which of the following statements are meaningful:

1. Peter is twice as tall as Hermann
2. Peter's temperature is 10% higher than Hermann's
3. Defect X is more severe than defect Y
4. Defect X is twice as severe as defect Y
5. The cost for correcting defect X is twice as high as the cost for correcting defect Y
6. The average temperature of city A (45 F) is twice as high as the average temperature of city B (90 F)
7. Project Milestone 3 (end of coding) took ten times longer than Project Milestone 0 (project start)
8. Coding took as long as requirements analysis

Exercise 4 (cont.)

Florida Tech

9. The length of Program A is 50.
10. The length of Program A is 50 executable statements.
11. Program A took 3 months to write.
12. Program A is twice as long as Program B.
13. Program A is 50 lines longer than Program B.
14. The cost of maintaining program A is twice that of maintaining Program B.
15. Program B is twice as maintainable as Program A.
16. Program A is more complex than Program B.