Software Architectures and Quality

Measurement Scales & Measurement in Software Engineering

Javier González Huerta

javier.gonzalez.huerta@bth.se

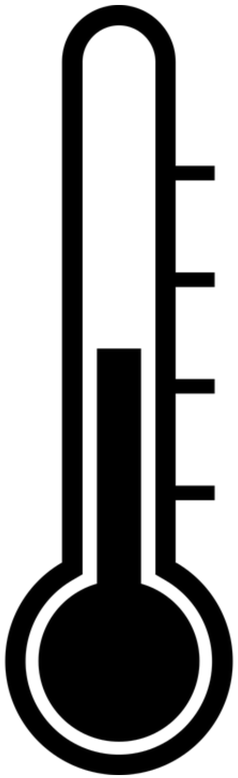
BLEKINGE TEKNISKA HÖGSKOLA · BTH ·

in real life

Objectives

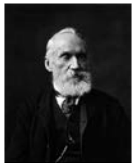
- Discuss and reflect about what is measurement and prediction
- Discuss and reflect about the different types of measurement scales, what can and cannot be done with each of them
- Discuss some examples on how to measure some qualities

What is the weather like today?



What is Measurement?

- Is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way that define them accordingly defined rules

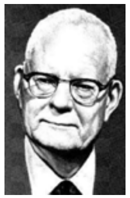


“When you can measure what you are speaking about, and express it in numbers, you know something about it”

Lord Kelvin (Mathematical Physicist and Engineer) 1883

“You can’t control what you can’t measure”

Tom DeMarco (Software Engineer) 1982



“In God we trust, all the others bring Data”

Edwards Deming, Data Scientist

Measurement Scales & Scale Types

- There are four main types of scales
  - Nominal
  - Ordinal
  - Interval
  - Ratio

Nominal Scales

- Nominal-level measurement are names or category labels
  - Categories should be exclusive
  - Categories must be discrete but also exhaustive
  - No relation between the labels
- Examples:
  - Type of Fruit: Apple, Pear, Banana
  - Weather: Sunny, Cloudy…
  - Gender: Male or Female
- Main Operations: Classification / Membership
- Operators: {= / !=}

Nominal Scales: What else can and cannot be done

- Central tendency: Mode (the most common element)
- The median does not make sense, since the concept of order does not exist
- The dispersion (as a mathematical concept) are also not applicable
- We can plot the values / use contingency tables

	Male	Female
Left handed	10	20
Right handed	49	28

Ordinal Scales

- Ordinal level also assign labels with information about an ordering of the classes or categories
  - There is not only category information but also rank information
  - We cannot say anything about the differences between the labels / levels
- Examples:
  - Weather: Scorching, hot, warm, tempered, chilly, cold, freezing, hell freezes over…
  - Order of finishing in the last blodomloppet race: 1^{st},2^{nd},… ,1030^{th}
- Main Operations: Comparison
- Operators: {Nominal} + {</>}

Ordinal Scales: What else can and cannot be done

- Central tendency: Median (middle-ranked) & Mode (most common)
- We cannot say anything about the differences between data points
  - Example: The difference in time between the first and second qualified in blodomloppet might not be the same than between 1030 and 1031
- In some cases we would be able to calculate the percentiles / interquartile ranges to measure dispersions

Interval Scales

- Interval scales captures information about the size of the intervals that separate the classes
  - Preserves order (as the ordinal)
  - Preserves differences but not ratios
  - Lack of an absolute 0 (the zero position is arbitrary)
- Examples:
  - Temperatures in Celsius / Fahrenheit
  - Dates
  - Scores in TOEFL
- Main Operations: Difference
- Operators: {Ordinal}+{+ / -}

Interval Scales: What else can and cannot be done

- Central tendency: Arithmetic Mean, Median (middle-ranked) & Mode (most common)
- Dispersion: Range and standard deviation
- The only statistical tools that should not be used are those that require the use of ratios, such as the coefficient of variation $C_v = \frac{\sigma}{\mu}$

Ratio Scales

- The ratio scale is the estimation of the ration between a magnitude of a continuous quantity and a unit magnitude of the same kind
- Ratio scales is a measurement mapping that preserves order, the size of intervals but also the ratios between entities
  - There is a true, non-arbitrary zero, representing the total lack of the attribute
- Examples:
  - Temperature in Kelvin
  - Length
  - Weight in pounds
- Main Operations: Ratio / Magnitude

Ratio Scales: What else can and cannot be done

- All statistical tools are applicable at this level
  - (taking into account the nature of the distributions)

And What For?

- Quality Analysis
- Architectural Evaluation / Assessment
- Quality Prediction

Quality Analysis

- Structured or unstructured examination of a particular quality
- Results in:
  - Quantitative scales (e.g., Memory Consumption = 164.23Mb on average)
  - Statements about the qualities using qualitative scales (e.g., good)
- Related to stakeholders' needs to be useful

Architecture Evaluation

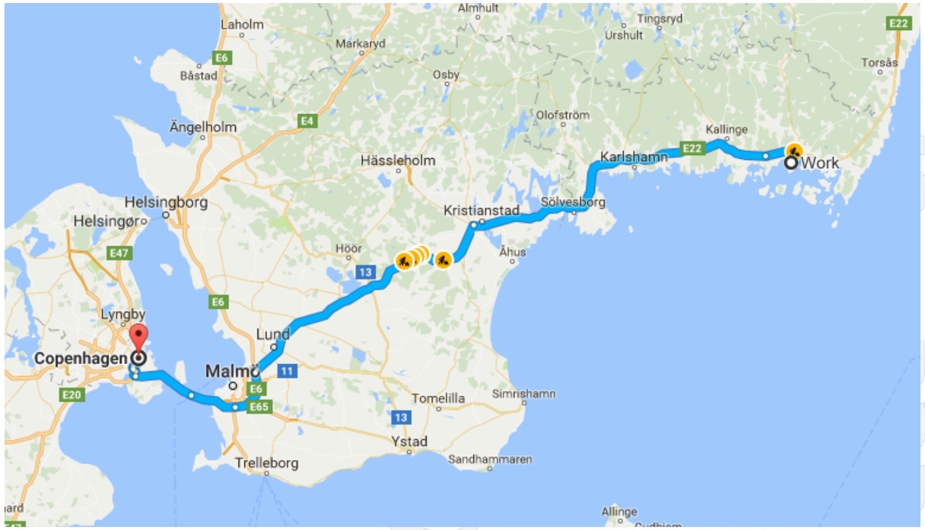
- Typically a qualitative examination of the architecture
  - In advance to the implementation
  - By an Internal or External team
  - Based on Evaluation methods: ATAM, SAAM, ALMA, etc.
- Results as conclusions in relation with stakeholders' needs/requirements
  - Typically qualitative, argumentative based mostly on participants' experience and knowledge (in assignment 4 we will try to have quantitative data too…)

Measurement for Prediction

- A prediction system consists of a mathematical model together with a set of prediction procedures for determining unknown parameters and interpreting the results
- In our context:
  - Structured examination of a particular quality that results in a statement about the implemented systems quality characteristics
  - Results in quantitative scales
    - E.g: Latency time= 46ms

Measurement for Prediction

- Suppose we want to predict the cost of a trip from here to Copenhagen
  - Entity we want to predict: Journey
  - Attribute: Cost



$$Cost = \frac{a\ * b}{c}$$

With:

- a being the distance in Kms between BTH and Copenhagen
- b being the cost per liter of diesel
- c distance in Kms we can travel with 1 liter of diesel

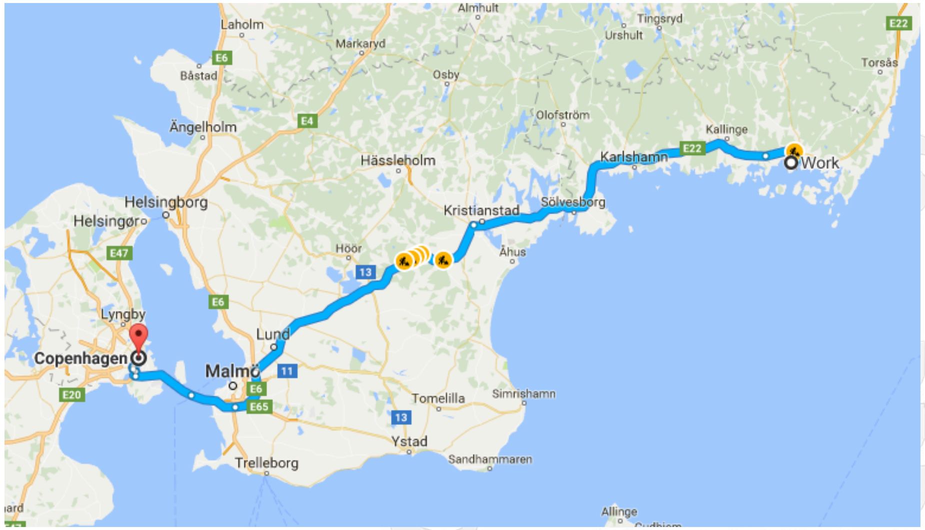
Measurement for Prediction

- Suppose we want to predict the cost of a trip from here to Copenhagen
  - Entity we want to predict: Journey
  - Attribute: Cost

$$Cost = \frac{a \ * b}{c}$$

Assumptions:

1. The distance between BTH and Copenhagen is ≃250kms (According to google maps)
2. The cost of a liter of diesel is 15kr
3. A car consumes ≃5lts/100kms: c=100/5; c=20



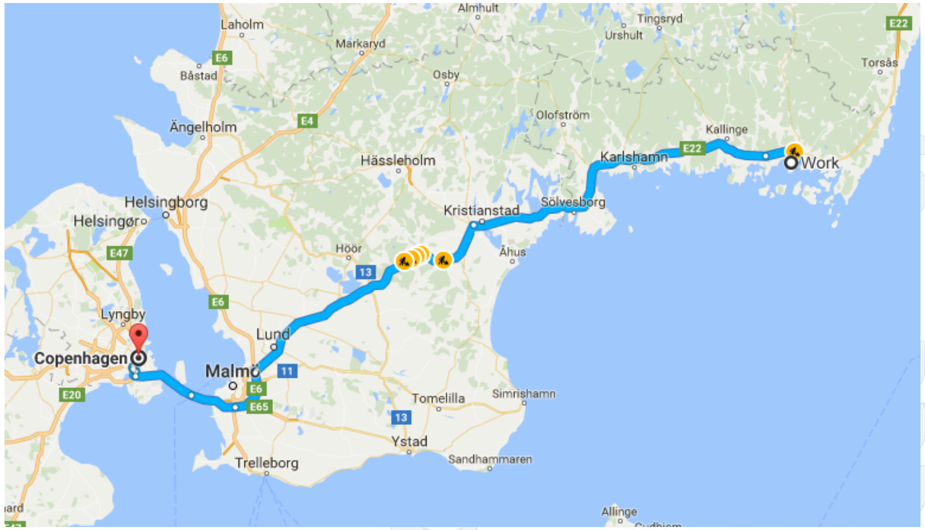
Measurement for Prediction

- Suppose we want to predict the cost of a trip from here to Copenhagen
  - Entity we want to predict: Journey
  - Attribute: Cost

$$Cost = \frac{a\ * b}{c} = 187.5$$

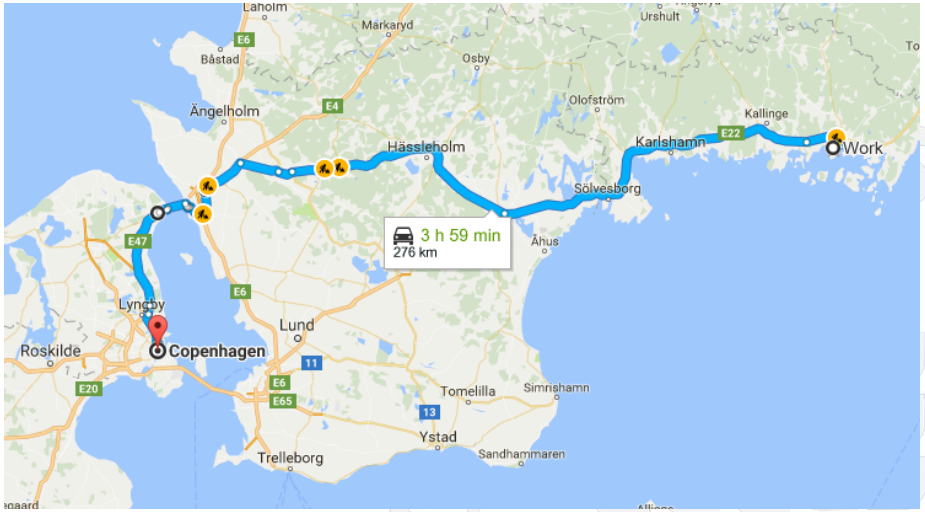
Assumptions:

1. The distance between BTH and Copenhagen is ≃250kms (According to google maps)
2. The cost of a liter of diesel is 15kr
3. A car consumes ≃5lts/100kms: c=100/5; c=20



Measurement for Prediction

- Accuracy?
  - Will depend on the assumptions



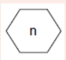
Direct vs Derived Measurement

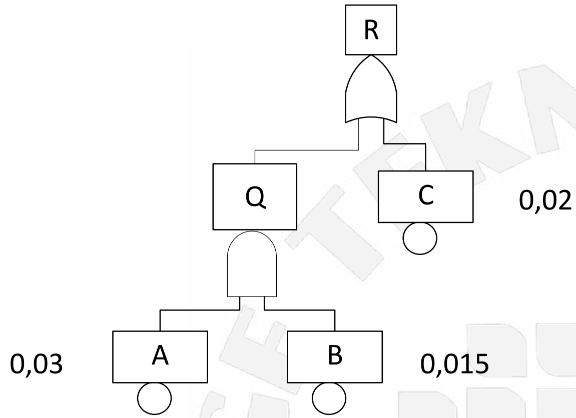
- After identifying entities and attributes we should define how to measure them:
  - **Direct Measurement –** Measuring an attribute of an entity does not involve any other attribute of the same or any other entity
  - **Derived (or Indirect) Measurement –** Measuring an attribute of an entity involves measuring other attribute of the same or any other entity

Some Derived Measures in SE

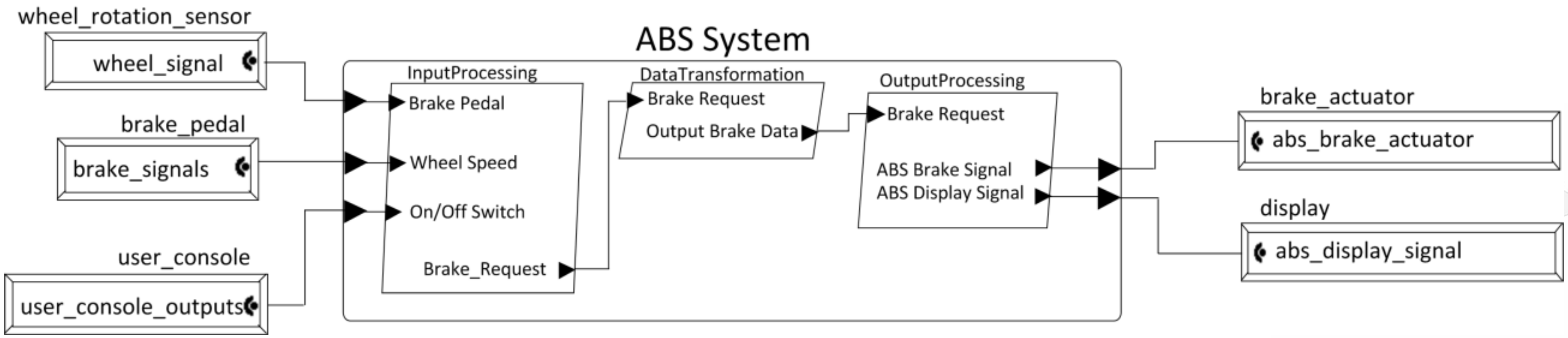
Programmer productivity	$\frac{LOC\ produced}{Persons\ months\ of\ effort}$
Module defect density	$\frac{\#of\ defects}{Module\ Size}$
Defect detection productivity	$\frac{\#\ of\ defects\ detected}{time}$
System availability	$\frac{MTBF}{MTBF + MTTR}$

Some examples in Software: Fault Tree Analysis

Element	Description	Probability Calculation
	Basic Event: it describes a failure, q basic initiating fault requiring no further development	P(A)
	AND: The AND Gate indicates that the output fault Q occurs if and only if all of the input faults of that gate occur simultaneously (A, B)	P(Q) = P(A) P(B)
	OR: The OR Gate indicates that the output fault Q occurs if one of the input faults of that gate occur (A, B)	P(Q) = P(A) + P(B)
 n	Combination: The output fault Q occurs only if n input faults occur simultaneously (A,B,C)	P(Q)=P(A)*P(B) + P(A)*P(C) + P (B)*P(C) If P(A)=P(B)=P(C) then P(Q)=3*(P(A))^2 (For n=2)



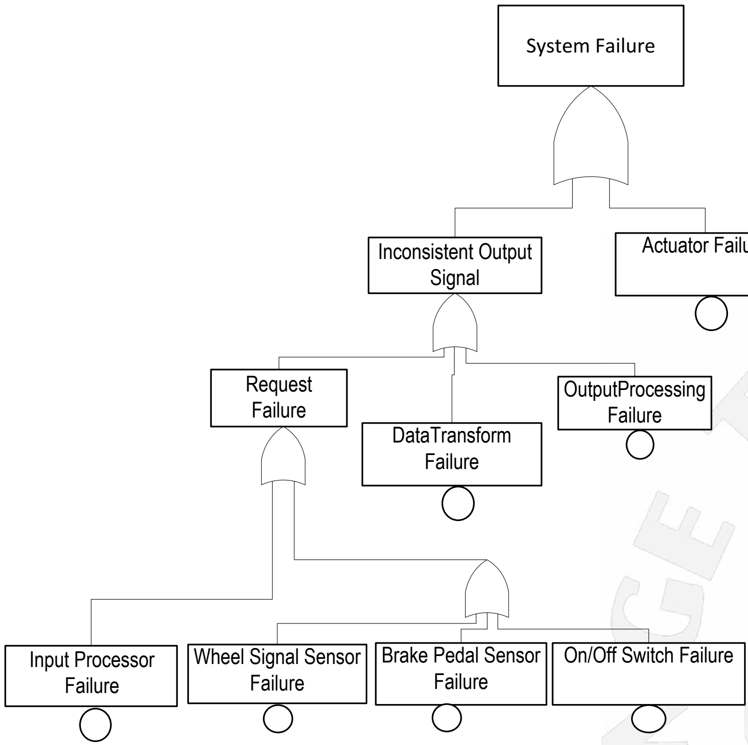
Some examples in Software: Fault Tree Analysis



Some examples in Software: Fault Tree Analysis

Assumptions:

Event	Probability
Sensor Failure	0,01
Input Processing Failure	0,015
Data Transformation Failure	0,015
Output Processing Failure	0,015
Actuator Failure	0,01
Comparator Failure(TMR)	0,01

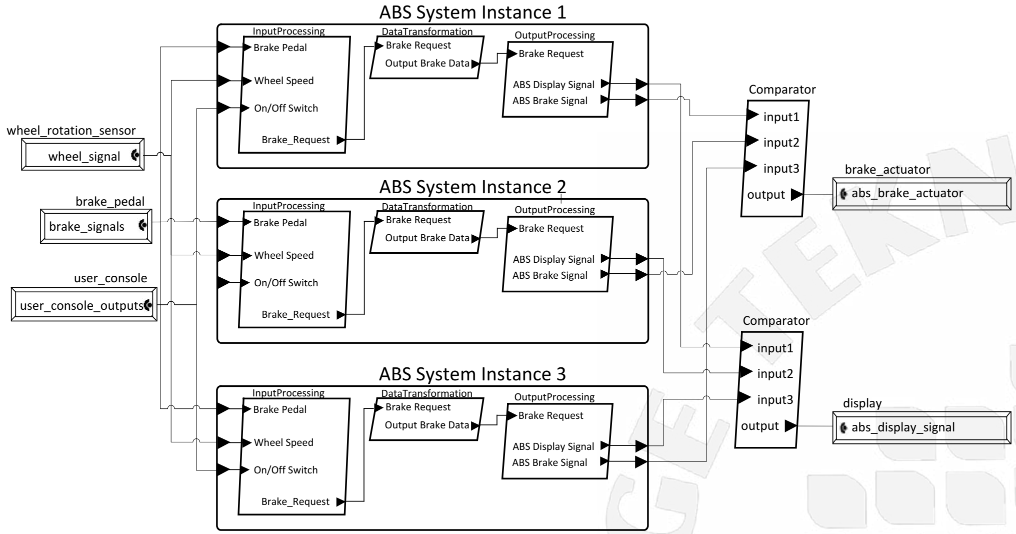


*P(System Fault) = (P(Actuator Failure) + P(Data Transform Failure)+P(Input Processing Failure) + P(Input Processing Failure) + 3*P(Sensor Failure))*

Some examples in Software: Fault Tree Analysis

Assumptions:

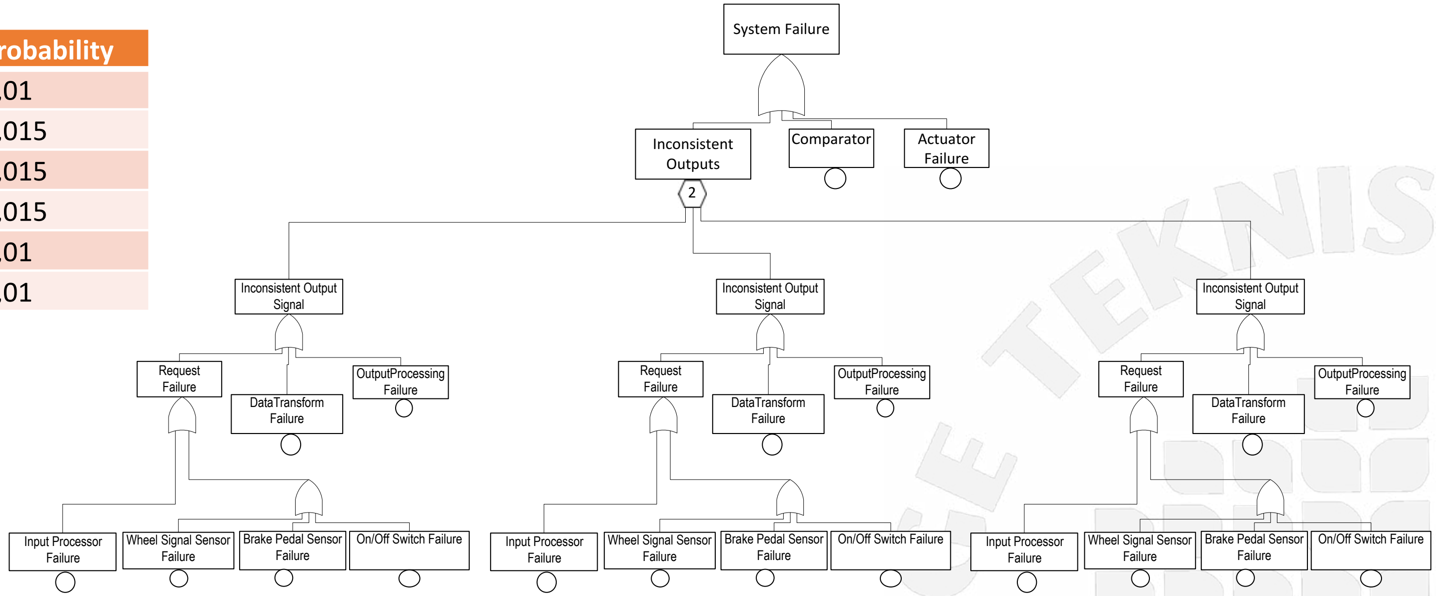
Event	Probability
Sensor Failure	0,01
Input Processing Failure	0,015
Data Transformation Failure	0,015
Output Processing Failure	0,015
Actuator Failure	0,01
Comparator Failure(TMR)	0,01



Some examples in Software: Fault Tree Analysis

Assumptions:

Event	Probability
Sensor Failure	0,01
Input Processing Failure	0,015
Data Transformation Failure	0,015
Output Processing Failure	0,015
Actuator Failure	0,01
Comparator Failure(TMR)	0,01



P(System Fault) = P(Actuator) + P(Comparator) + 3*(P(Output Processing Failure) + P(Data Transform Failure)+P(Input Processing Failure) + 3*P(Sensor Failure)))^2

Availability

$$Availabiltiy = \frac{MTBF}{MTBF + MTTR}$$

- MTBF can be assessed as a result of the fault tree analysis

$$P(T) = e^{-T/MTBF}$$

- MTTR the time it takes to become online again

Modifiability

- Define some scenario of all types of relevant future / past changes
- Grade each scenario based on its probability / frequency
- Perform impact analysis for each scenario
  - Estimate (or search in the archives) the volume of code changed / modified / deleted
- With this information you can calculate the effort to implement each change
- Summarize by calculating the weighted average

Modifiability

Modifiability scenario	Expected Frequency	Change Volume
Change the encryption protocol	2	300LOC
Add new fields to the GUI forms	25	10LOC
Support new export file format	2	40LOC

Modifiability = (300 * 2 + 10 *25 + 2 *40) / (2 + 25 + 2) = 32 LOC / Modification

Interoperability

- Relevant types of information exchanges needed
  - Normally this would be functional requirements in the requirements spec
- Grade each exchange based on its value to customer or importance
- Evaluate each exchange by predict/calculate:
  - – The probability for successful information exchange

Interoperability

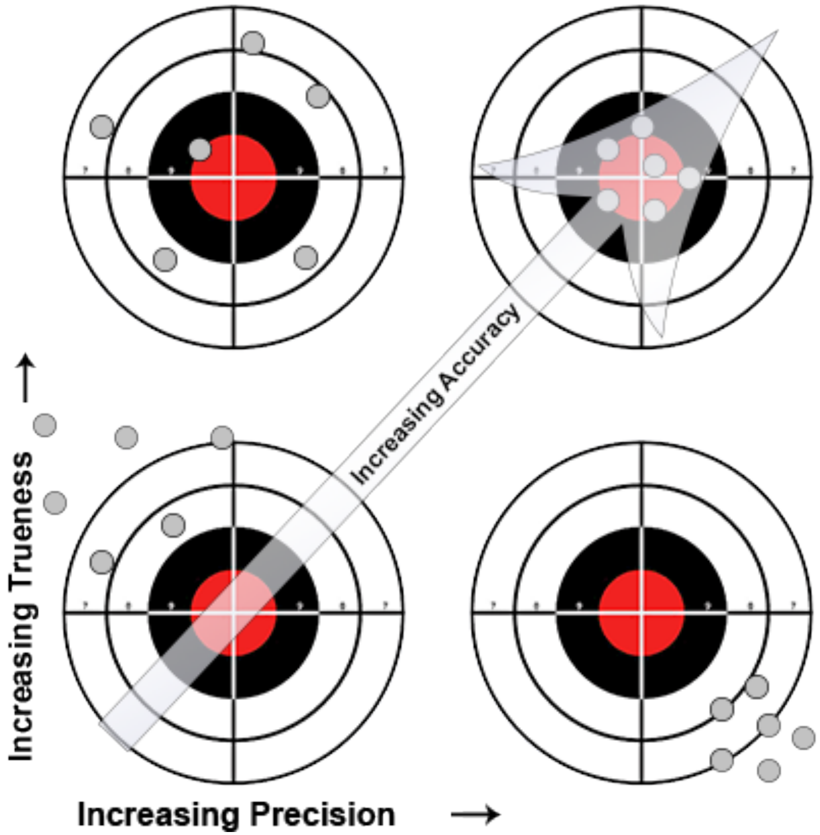
Modifiability scenario	Expected Frequency	Information Correctly Imported
Import bibliographic information from Google Scholar to Mendeley	55	35%
Import bibliographic information from Springer to Mendeley	25	60%
Import bibliographic information from IEEE to Mendeley	25	90%
Import bibliographic information from Journal page to Mendeley	5	100%

Interoperability= (55 *35 + 25 *60 + 25 * 90 + 5 * 100) / 100 = 61.75

Repeatability and Reproducibility

- **Repeatability** is the variation in measurements taken by a single person or instrument on the same item, under the same conditions
- **Reproducibility:** The ability of an instrument, used by multiple operators, to consistently reproduce the same measurement of the same entity, under the same conditions.

Accuracy, Precision and Trueness



Software Architectures and Quality

Measurement Scales & Measurement in Software Engineering

Javier González Huerta

javier.gonzalez.huerta@bth.se

BLEKINGE TEKNISKA HÖGSKOLA · BTH ·

in real life