Course 2

Software Quality Scales, Evaluations and Tools

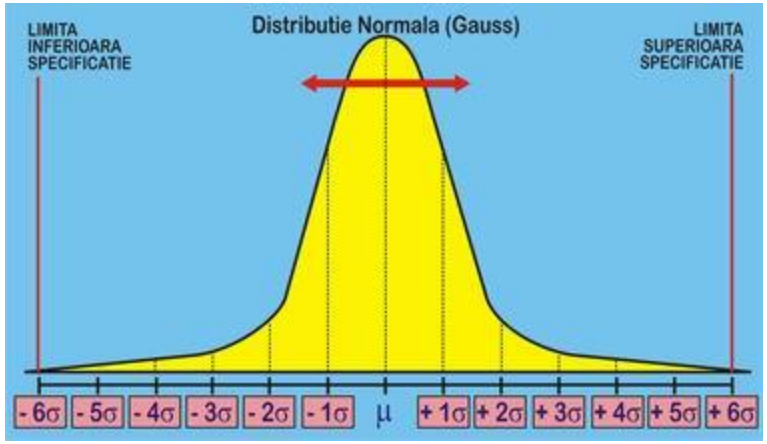
Measurements

- Nominal scale
- Ordinal scale
- Interval scale
- Ratio scale

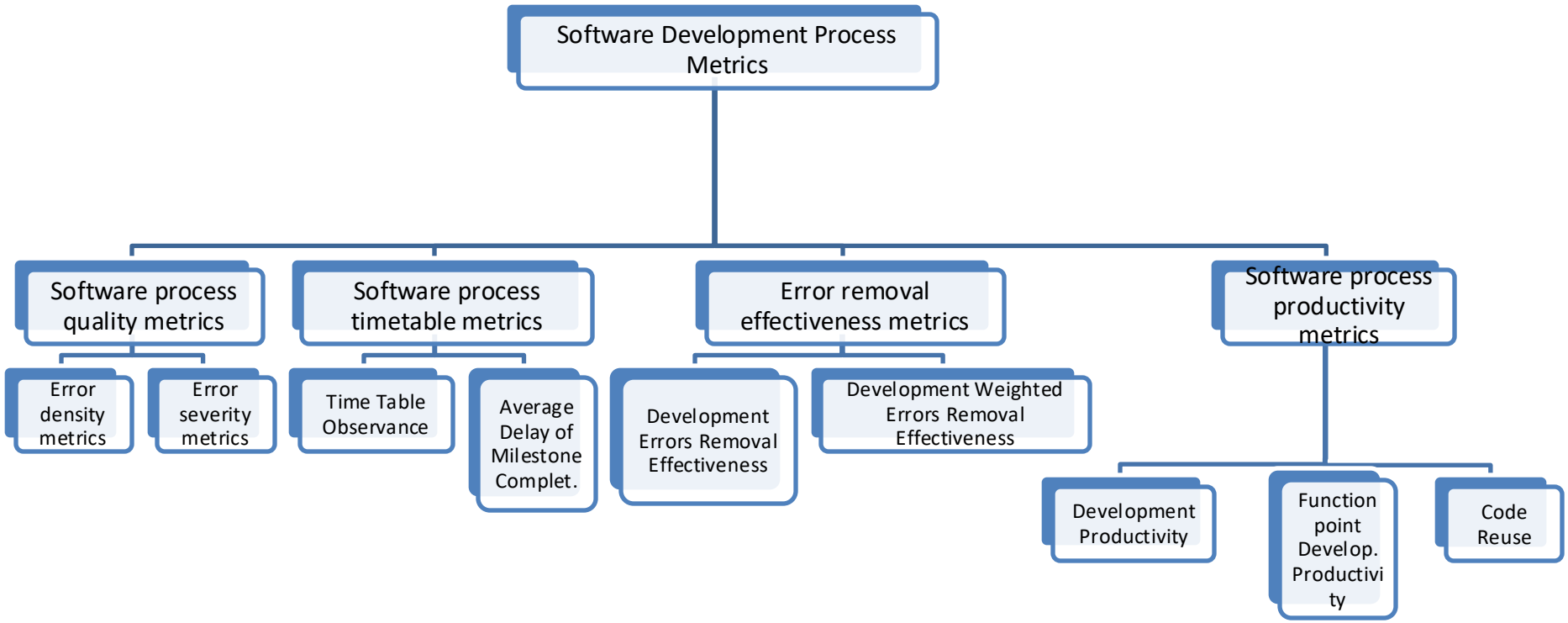
- https://github.com/acmsigsoft/EmpiricalStandards/blob/master/docs/supplements/InformationVisualization.md

Basic measures

- Ratio
- Proportion
- Percentage
- Rate
- Six sigma



- Quality level
- Motorola, 1986
-Value = defect rate = 3.4 defective parts / milion


Software Development Process Metrics
Software process quality metrics
Software process timetable metrics
Error removal effectiveness metrics
Software process productivity metrics
Error density metrics
Error severity metrics
Time Table Observance
Average Delay of Milestone Complet.
Development Errors Removal Effectiveness
Development Weighted Errors Removal Effectiveness
Development Productivity
Function point Develop. Productivity
Code Reuse

Example 1

- Error severity metrics:

  ASCE (Average Severity of Code Errors)

$$ASCE = \frac{\text{weighted code errors detected}}{\text{number of code errors detected}}$$

Data collected from code inspection and testing reports

Example 2 (SQ+PM)

- Technical debt = describes the consequences of software development actions that intentionally or unintentionally prioritize client value and/or project constraints such as delivery deadlines, over more technical implementation and design consideration [cit]

- Signs for:
  - Code smells
  - High complexity
  - Product bugs
  - Coding style

- SonarQube https://docs.sonarqube.org/display/SONARQUBE52/Technical+Debt

Implementation of SQ Metrics

- Definition of SQ metrics
  - Relevant and adequate – teams, depart., projects

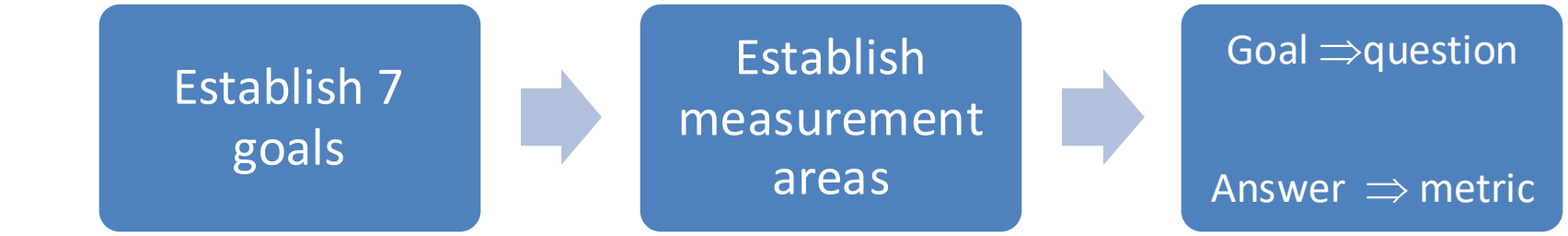
1. Definition of attributes to be measured
2. Definition of the metrics
3. Determination of comparative target values
4. Reporting and data collection method

- Application – management
  - Process [Galin]
- Statistical analysis of collected data

Agile

Example 1. Motorola

- Daskalantonakis, M. K., "A Practical View of Software Measurement and Implementation Experiences Within Motorola (1001–1004)," *IEEE Transactions on Software Engineering, Vol. 18, No. 11 (November 1992): 998–1010. Copyright © IEEE, 1992*



Project planning, software productivity

Example 2. Hewlett-Packard

- Grady, R. B., and D. L. Caswell*, Software Metrics: Establishing A Company-Wide Program, pp. 225–226. Englewood Cliffs, N.J.: Prentice-Hall*

- Primitive + computed metrics

  12 metrics

VISUALIZATION TOOLS

7 Basic Quality Tools

- Ishikawa (1989)
- Quality control
- statistical tools for process and quality control
  - project and organization level
  - useful for project leaders and process experts.
- do NOT provide specific information to software developers on how to improve the quality of their designs or implementation.

1. Checklist/ check sheet

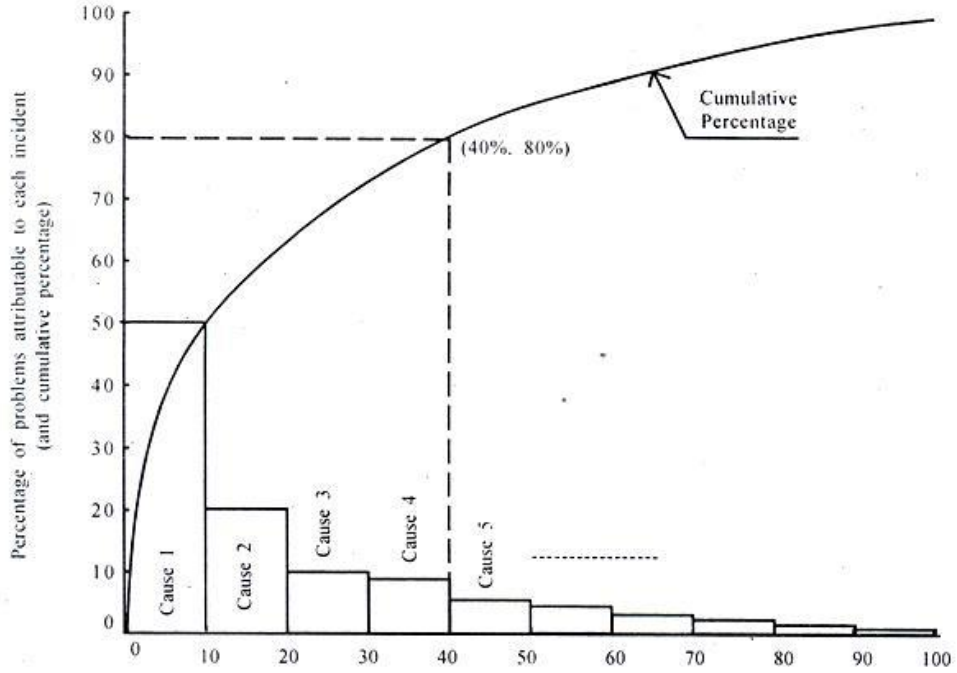
- Gathering and arranging data + confirmation
- Types:
  - design review checklist
  - code inspection checklist
  - Example: Requirement checklist

When to use it?
•When data observed and collected repeatedly
•When collecting data from a production process.

How to use it?
1. Decide what event or problem will be observed. Develop operational definitions.
2. Decide when data will be collected and for how long.
…

2. Pareto Diagram

- X axis = defect cause
- Y axis = defect count
- Pareto analysis: 80-20%



2. Pareto Diagram

When to use it?

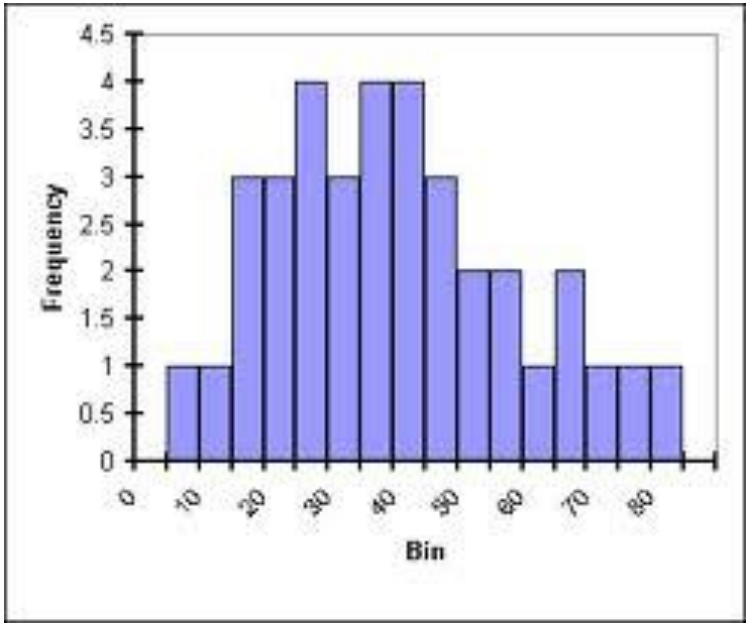
- When analyzing data about the frequency of problems or causes in a process.
- When analyzing causes by looking at their specific components.
- When communicating with others about your data.

How to use it?

1. Decide categories to group items.
2. Decide measurement: frequency, quantity, cost and time.
3. Decide period of time
4. Collect the data, recording the category each time.
5. Subtotal the measurements for each category.
6. Determine the appropriate scale for the measurements you have collected. The maximum value will be the largest subtotal from step 5. Mark the scale on the left side of the chart.
7. Construct and label bars for each category. Place the tallest at the far left, and so on.

3. Histogram

- graphic representation of frequency
- X-axis lists the unit intervals of a parameter
- Y-axis contains the frequency counts.
- order of X variable, (Pareto diagram - order of the frequency counts)
- show the distribution characteristics of a parameter



3. Histogram

How to use it?

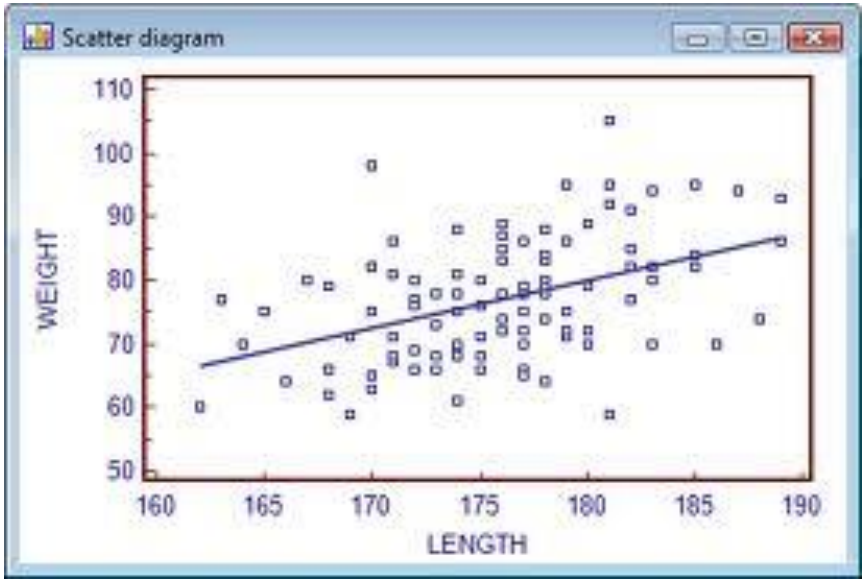
1. Collect at least 50 consecutive data points from a process.
2. Analyze shape and meaning

When to use it?

- When the data are numerical.
- When you want to see the shape of the data's distribution, especially when determining whether the output of a process is distributed approximately normally.
- When analyzing whether a process can meet the customer's requirements.

4. Scatter diagram

- the relationship of two interval variables
- X-axis = independent variable
- Y-axis = dependent variable
- Each point in a scatter diagram represents an observation of both the dependent and independent variables
- help data-based decision making



4. Scatter diagram

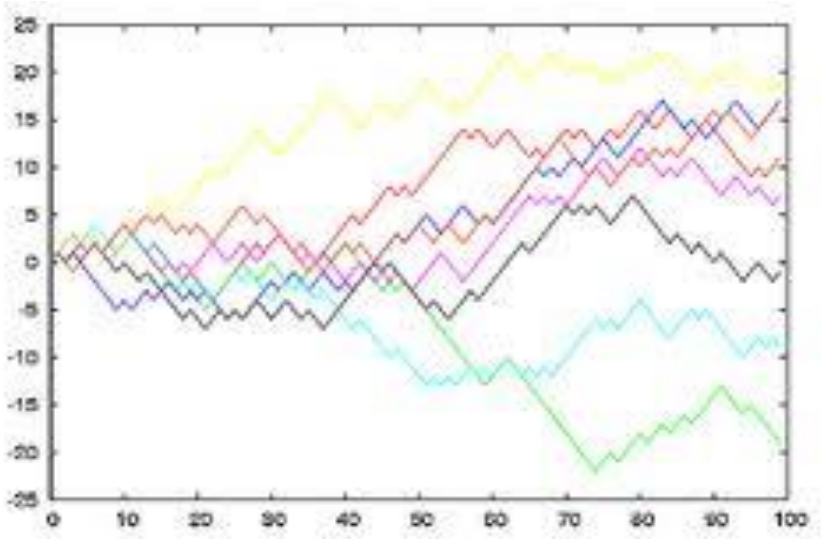
How to use it?

1. Collect pairs of data where a relationship is suspected.
2. Look at the pattern of points to see if a relationship is obvious. If the data clearly form a line or a curve, you may stop. The variables are correlated.
3. You may wish to use regression or correlation analysis now.

- **When to use it?**
- When you have paired numerical data.
- When your dependent variable may have multiple values for each value of your independent variable.
- When trying to determine whether the two variables are related

5. Run chart

- tracks the performance of parameter over time
- X-axis = time
- Y-axis = value of the parameter.
- used for trend analysis
- pie chart, bar graph, circle graph [Ishikawa]



5. Run chart

When to use it?

•When data come from several sources or conditions,

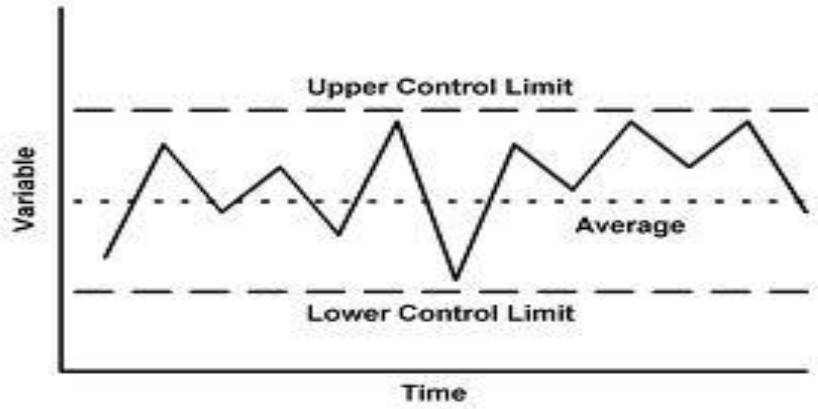
• when following the data in time / space

How to use it?

1. Consider which information about the sources of the data might have an effect on the results
2. Analyze the subsets of stratified data separately: draw quadrants, count points and determine the critical value for each set of data

6. Control chart

- advanced form of a run chart
- central line, a pair of control limits
- X-axis = real time.
- If all values of the parameter are within the control limits and show no particular tendency => controlled state.
- If values outside the control limits or indicate a trend => out of control



6. Control chart

When to use it?

•When controlling ongoing processes by finding and correcting problems as they occur.

•When predicting the expected range of outcomes from a process.

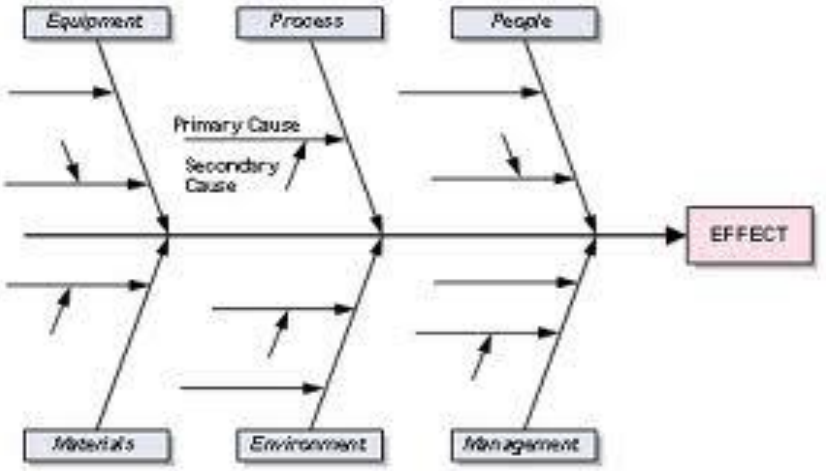
•When determining whether your quality improvement project make fundamental changes to the process.

How to use it?

1. Choose the appropriate control chart for your data.
2. Determine the appropriate time period for collecting and plotting data.
3. Collect data, construct your chart and analyze the data.
4. Look for “out-of-control signals” on the control chart. When one is identified, mark it on the chart and investigate the cause

7. Cause-and-effect diagram

- (fishbone diagram)
- shows the relationship between a quality characteristic and factors that affect that characteristic
- identifies all causal factors of a quality characteristic in one chart



7. Cause-and-effect diagram

When to use it?

•When identifying possible causes for a problem.

•Especially when a team's thinking tends to fall into ruts.

• Agile

How to use it?

1. Agree on a problem statement (effect)
2. Brainstorm the major categories of causes of the problem. If this is difficult use generic headings
3. Write the categories of causes as branches from the main arrow.
4. Brainstorm all the possible causes of the problem.

Recommendation for project

- Use a type of diagram for software quality evaluation