# SOFTWARE QUALITY

## CPTS 583

Quality Metrics and Measurement (II)

-- *Software metrics, quality metrics and measurement*

# Outline

- Software Metrics
  - Project/product/process metrics
- Software Quality Metrics
  - Product quality metrics
  - Process quality metrics
  - Maintenance Metrics

# Why Measure Software?

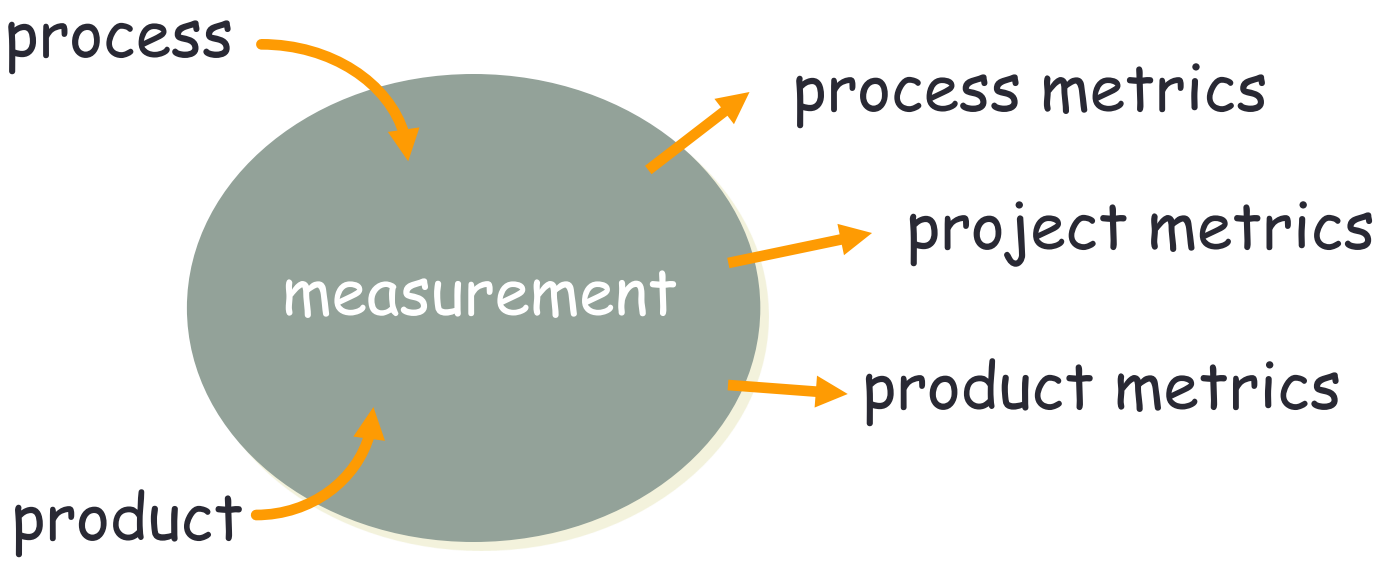
| | |
|---|---|
| ***Estimate cost and effort*** | measure correlation between specifications and final product |
| ***Improve productivity*** | measure value and cost of software |
| ***Improve software quality*** | measure usability, efficiency, maintainability ... |
| ***Improve reliability*** | measure mean time to failure, etc. |
| ***Evaluate methods and tools*** | measure productivity, quality, reliability ... |

*"You cannot control what you cannot measure" — De Marco, 1982*
*"What is not measurable, make measurable" — Galileo*

# Terminology

- **Measure**: Quantitative indication of the extent, amount, dimension, or size of some attribute of a product or process. A single data point
- **Metrics**: The degree to which a system, component, or process possesses a given attribute. Relates several measures (e.g. average number of errors found per person hour)
- **Indicators**: A combination of metrics that provides insight into the software process, project or product
- **Direct Metrics**: Immediately measurable attributes (e.g. line of code, execution speed, defects reported)
- **Indirect Metrics**: Aspects that are not immediately quantifiable (e.g. functionality, quality, reliability)
- **Errors**: issues found by the practitioners during software development
- **Defects**: issues found by the customers after release

# Scope of software measurement




"*Not everything that can be counted counts, and not everything that counts can be counted.*" - Einstein

# Scope of software measurement

- Process

  Measure the efficacy of processes. What works, what doesn't.

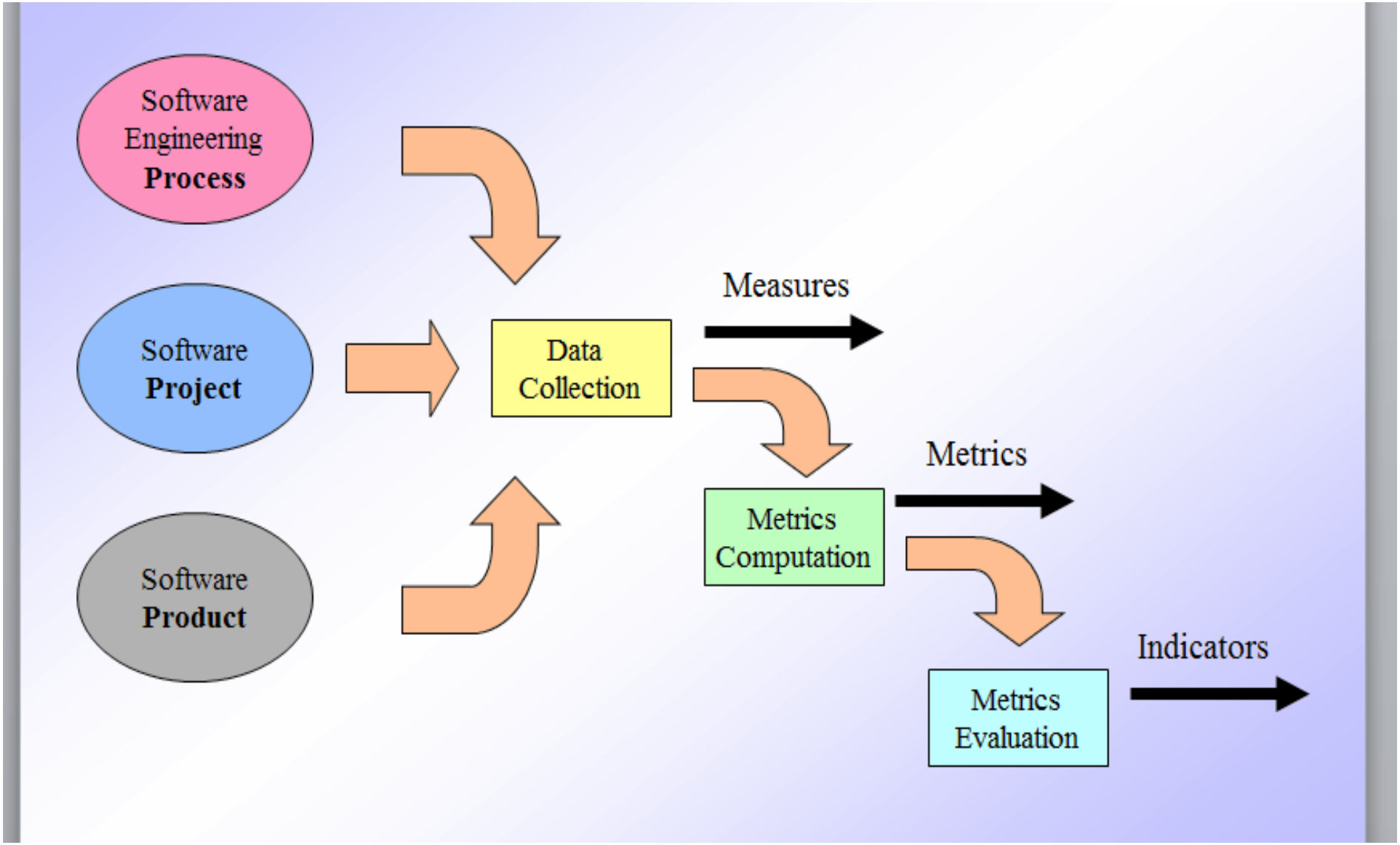
- Project

  Assess the status of projects. Track risk. Identify problem areas. Adjust work flow.

- Product

  Measure predefined product attributes (generally related to ISO9126 Software Characteristics)

# Scope of software measurement



# Project Metrics

- Used by a project manager and software team to adapt project work flow and technical activities.
- Tactical and short-term
- Purpose:
  - Minimize the development schedule by making the necessary adjustments to avoid delays and mitigate problems/risks
  - Assess product cost on an ongoing basis

# Project Metrics

- Effort or time per software engineering (SE) task
- Errors uncovered per review hour
- Scheduled vs. actual milestone dates
- Number of changes and their characteristics
- Distribution of effort on SE tasks
- Number of software developers
- Key performance indicator (KPI)
- Productivity

# Process Metrics

- Focus on quality achieved as a consequence of a repeatable or managed process.
- Strategic and Long Term.
- Example: Defect Removal Efficiency (DRE).
  - Relationship between errors (E) and defects (D).
  - The ideal is a DRE of 1:

$$DRE = E/(E+D)$$

- More example metrics:
  - Pattern of defect arrival
  - Response time for bug fixes

# Process Metrics

- Involves analysis of the way a product is developed
- What lifecycle do we use?
- What deliverables are produced?
- How are they analysed?
- How can the process help to produce products faster?
- How can the process help to produce better products?

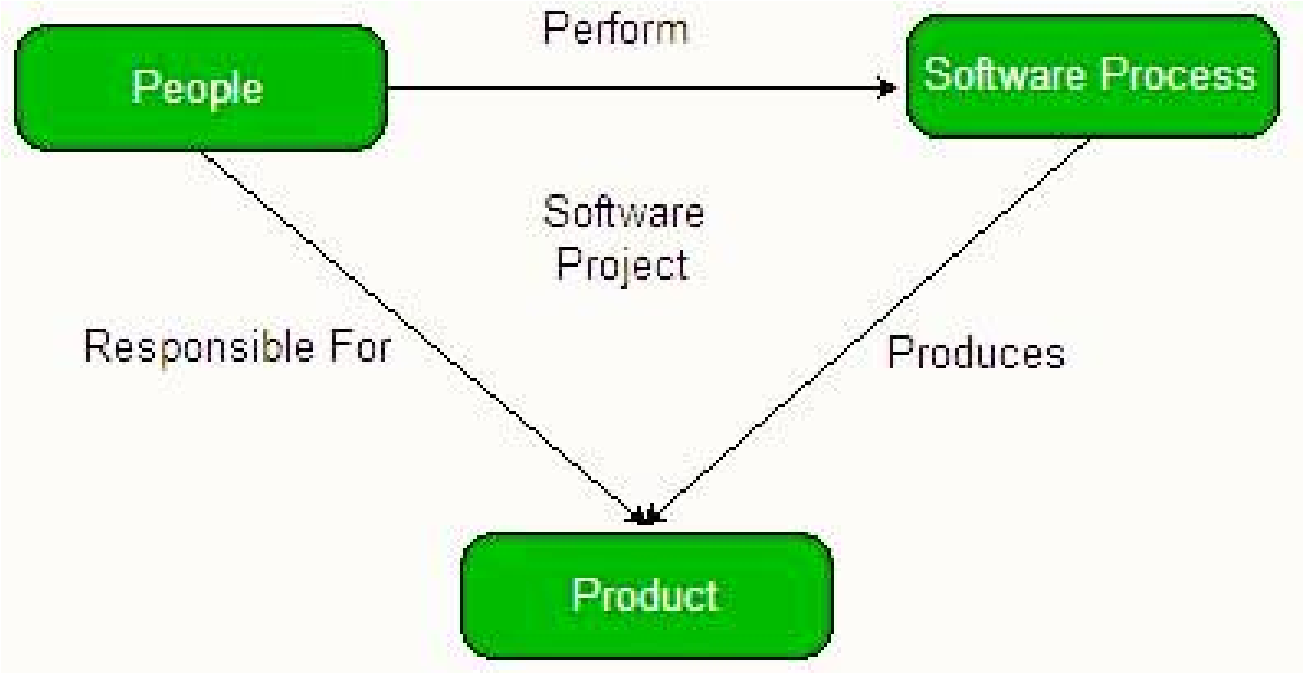
# Process Metrics

- Statistical Software Process Improvement (**SSPI**)
  - **Error/Defect Categorization and Analysis**

1. Categorize errors/defects
   - ↳ 2. Count errors/defects in each category
     - ↳ 3. Record costs of correcting each error/defect
       - ↳ 4. Analyze the data to find most costly categories
         - ↳ 5. Modify the process to reduce the outstanding costs

# Product Metrics

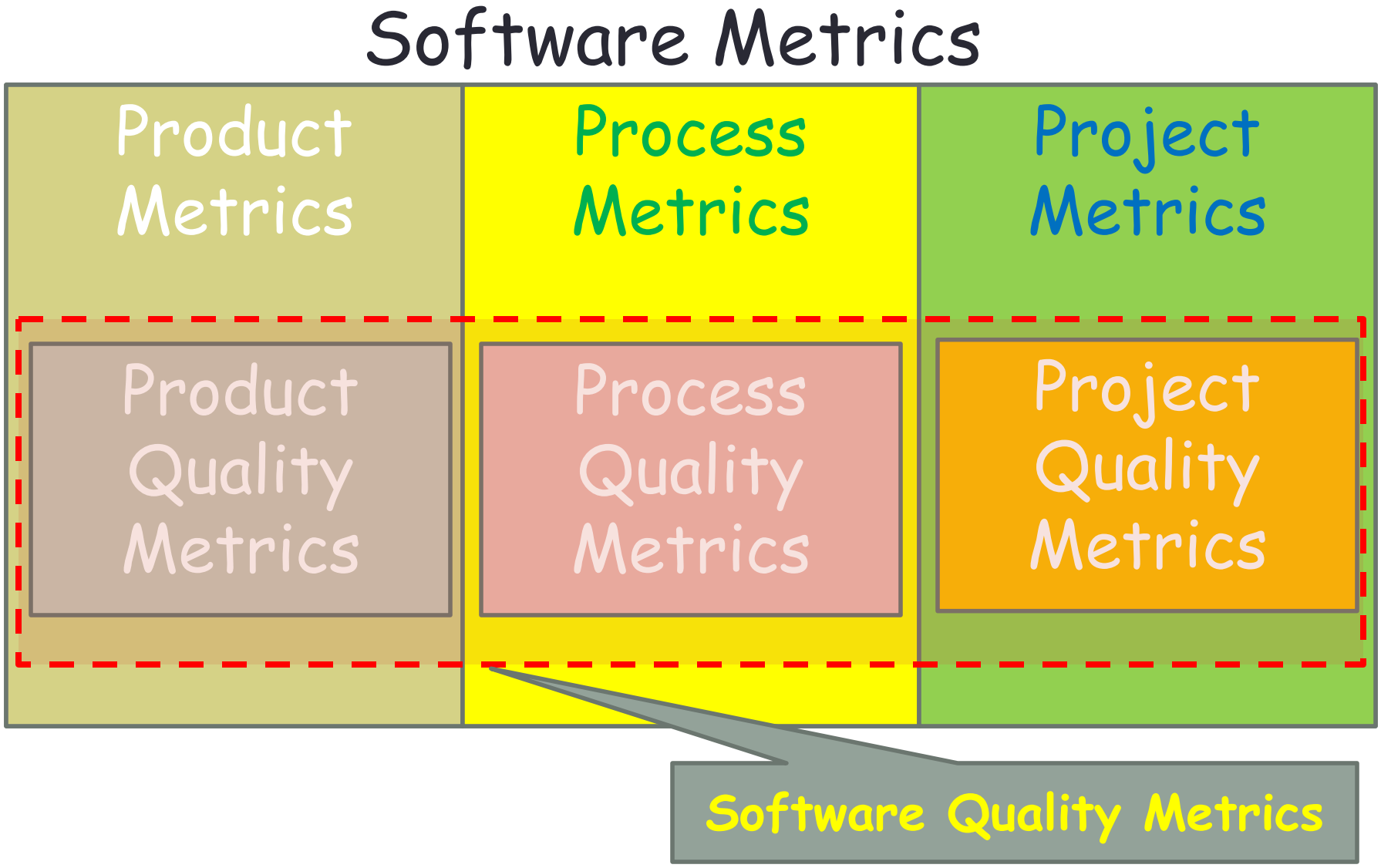
- Product refers to the actual software system, documentation and other deliverables
- Focus on the quality of deliverables
- Product metrics are combined across several projects to produce process metrics



# Product Metrics

- Code metrics
  - Size / complexity (LOC, FP)
- Functionality offered
- Cost
- Various Quality Attributes
- Measures of the Analysis Model
- Complexity of the Design Model
  1. Internal algorithmic complexity
  2. Architectural complexity
  3. Data flow complexity

# Quality Metrics as a Subset



# Software Quality Metrics - Motives

- How large was the product?
- What was the overall productivity of the software engineering group on the product?
- How many bugs were found before it was released?
- How many bugs did the customers find in the first three months after release?
- Was the overall quality better or worse than previous products?

# Software Quality Metrics - Merits


- Objective assessments as to whether quality requirements are being met can be made during development
- A quantitative assessment of quality can provide the basis for decisions regarding the software's fitness for use
- The effectiveness of the software development process can be objectively assessed

# Software Quality Metrics - Definition

## IEEE definitions of software quality metrics

(1) *A quantitative measure* of the degree to which an item possesses a given quality attribute.

(2) *A function* whose inputs are software data and whose output is a single numerical value that can be interpreted as the degree to which the software possesses a given quality attribute.

# Software Quality Metrics - Objectives

**1. Facilitate management control, planning and managerial intervention.**
Based on:

1) Deviations of actual from planned performance.
2) Deviations of actual timetable and budget performance from planned.

**2. Identify situations for development or maintenance process improvement (preventive or corrective actions).**

- Based on Accumulation of metrics information regarding the performance of teams, units, etc.

# Software Quality Metrics - Requirements

**General requirements**

- Relevant
- Valid
- Reliable
- Comprehensive
- Mutually exclusive

**Operative requirements**

- Easy and simple
- Does not require independent data collection
- Immune to biased interventions by interested parties

# Software Quality Metrics - Categorization

- Lifecycle based categorization
  - Process quality metrics
  - Product quality metrics

- Measurement objective based categorization
  - Cost (timeline/timetable)
  - Effectiveness (e.g., error removal)
  - Productivity

# Quality Measurement Guidelines

- ✓Apply common sense in metrics interpretation
- ✓Normalize if necessary

- ✓Offer feedbacks on measures and metrics workers.
- ✓Appraise software quality, NOT individuals
- ✓Never use metrics to threaten individuals or teams

- ✓Set clear goals and metrics for realizing the goals
- ✓Don't consider metrics "negative"; focus on process improvement

- ✓Be inclusive with metrics

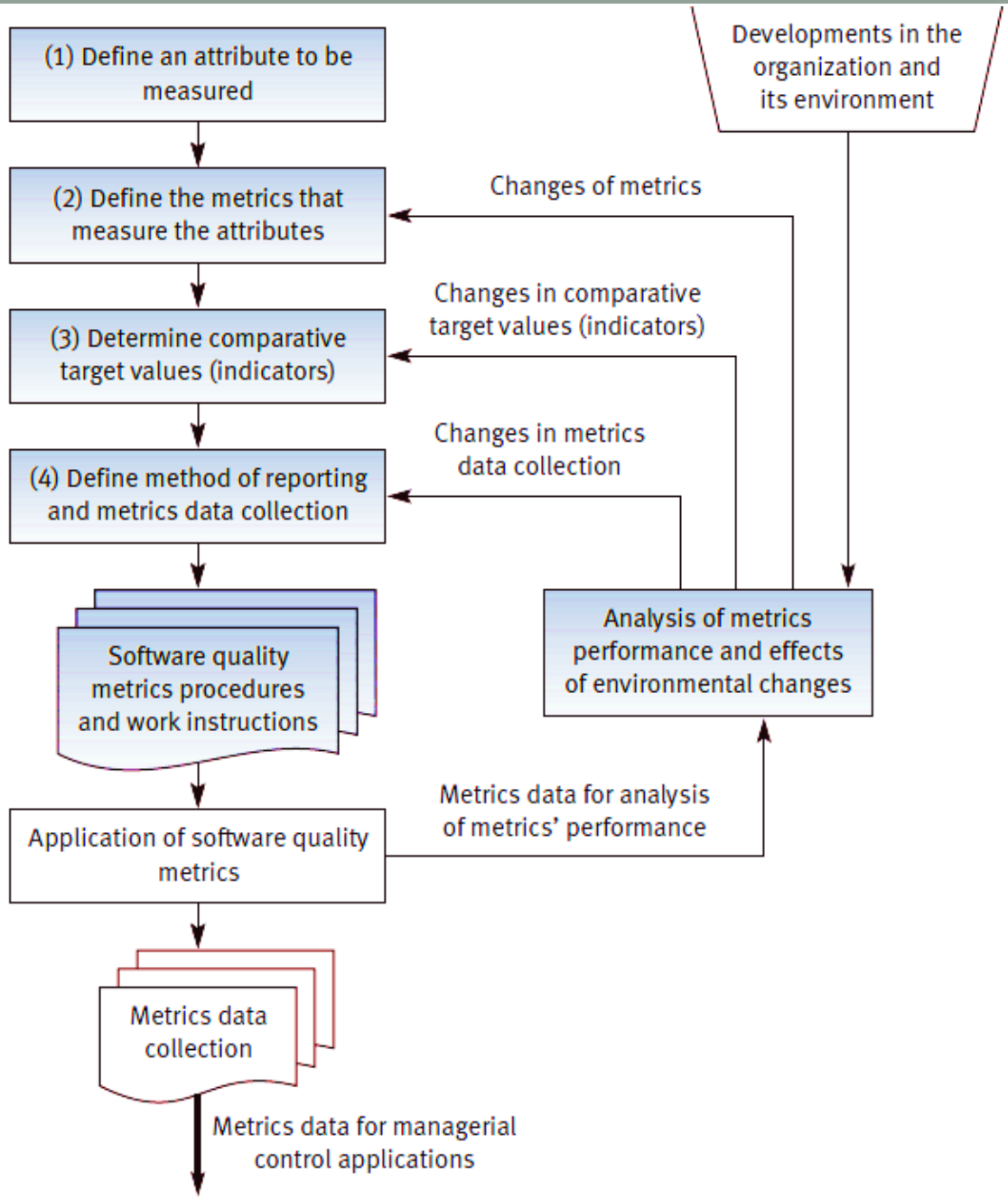
# Documenting Software Quality Metrics

| Item | Description |
|---|---|
| Name | Number of defects detected in selected modules |
| Costs | Minimal: data can be obtained from bug-tracking tool |
| Target Value | 5 |
| Tools | Spreadsheet |
| Application | Metric is used for relative comparison to values obtained for other modules |
| Data Items | Count of defects detected at code inspections |
| Computation | Sum number of defects reported against specific modules |

# Quality Metrics Procedure

1. Define measurement object
2. Determine metrics
3. Set targets
4. Decide the metrics collection method
5. Apply metrics

✓Dealing with changes



# Quality Metrics Constraints

* **Budget** constraints in allocating the necessary resources.

* **Human factors** especially opposition of employees to evaluation of their activities.

* **Validity** uncertainty regarding the data's, partial and biased reporting.

# Metrics Data Analysis

- Results need to be analyzed within the context of the project's overall software quality requirements
- Any metrics that fall outside of their respective targets should be identified for further analysis
- Assess the statistical significance of the metrics to the quality factors they represent

# Summary

- Software Metrics: project, process, product level
  - Motivation
  - Terminologies
  - Usefulness
  - Example metrics
  - Methodology
- Software Quality Metrics
  - Subset of software metrics
  - Motives, merits, definition, objective
  - Requirements, categorization, documentation
  - Procedure
  - Constraints
  - Analysis