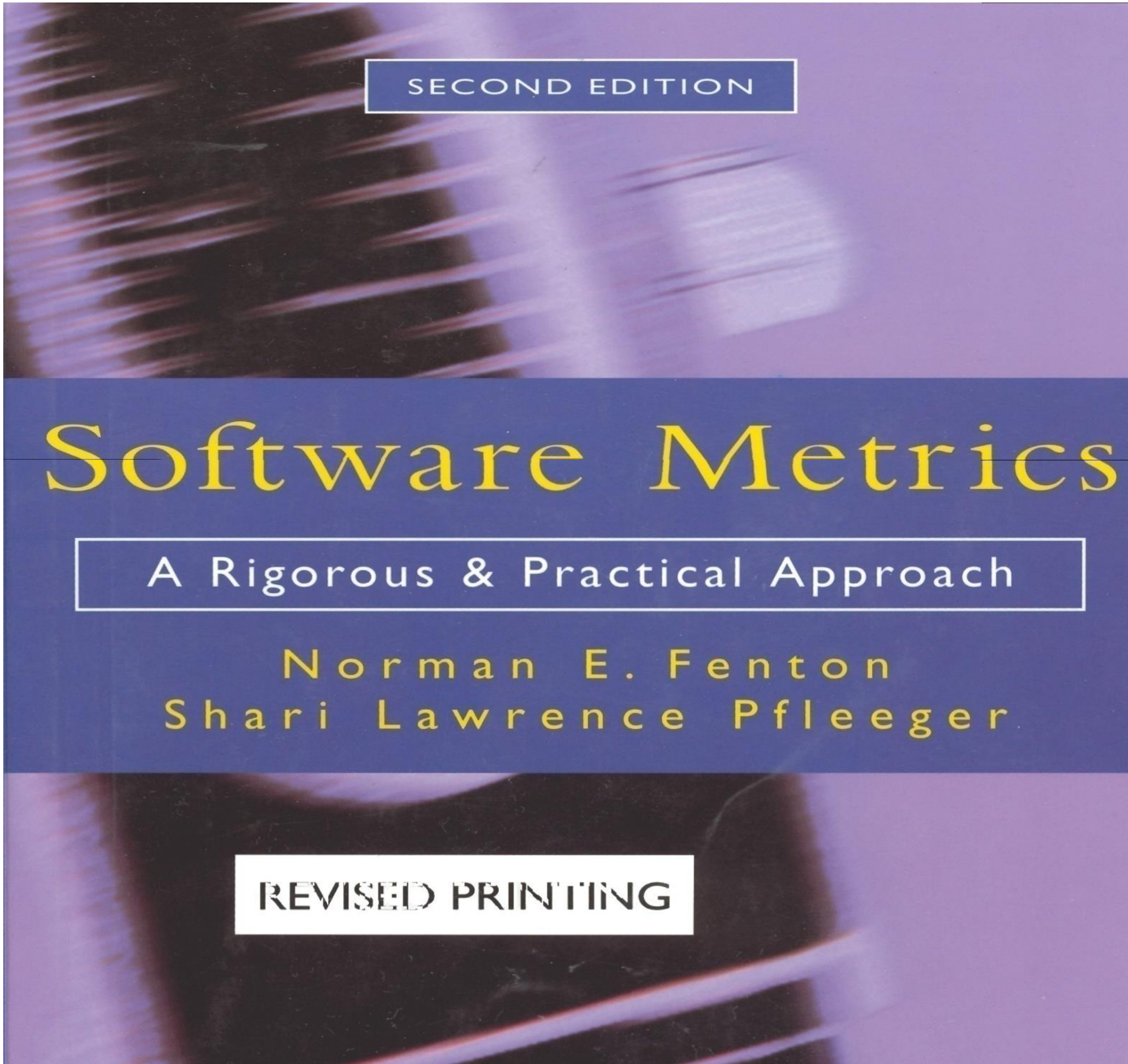
Software Metrics

Fundamentals of Measurements and Experimentation

SOFTWARE METRICS

TEXT BOOK

- Title : ***Software Metrics***
- Author : ***Norman E. Fenton and Share Lawrence Pfleeger***
- Publisher : ***International Thomson Computer Press, 2003.***
- Edition : ***Thomson Delmar 2nd edition.***

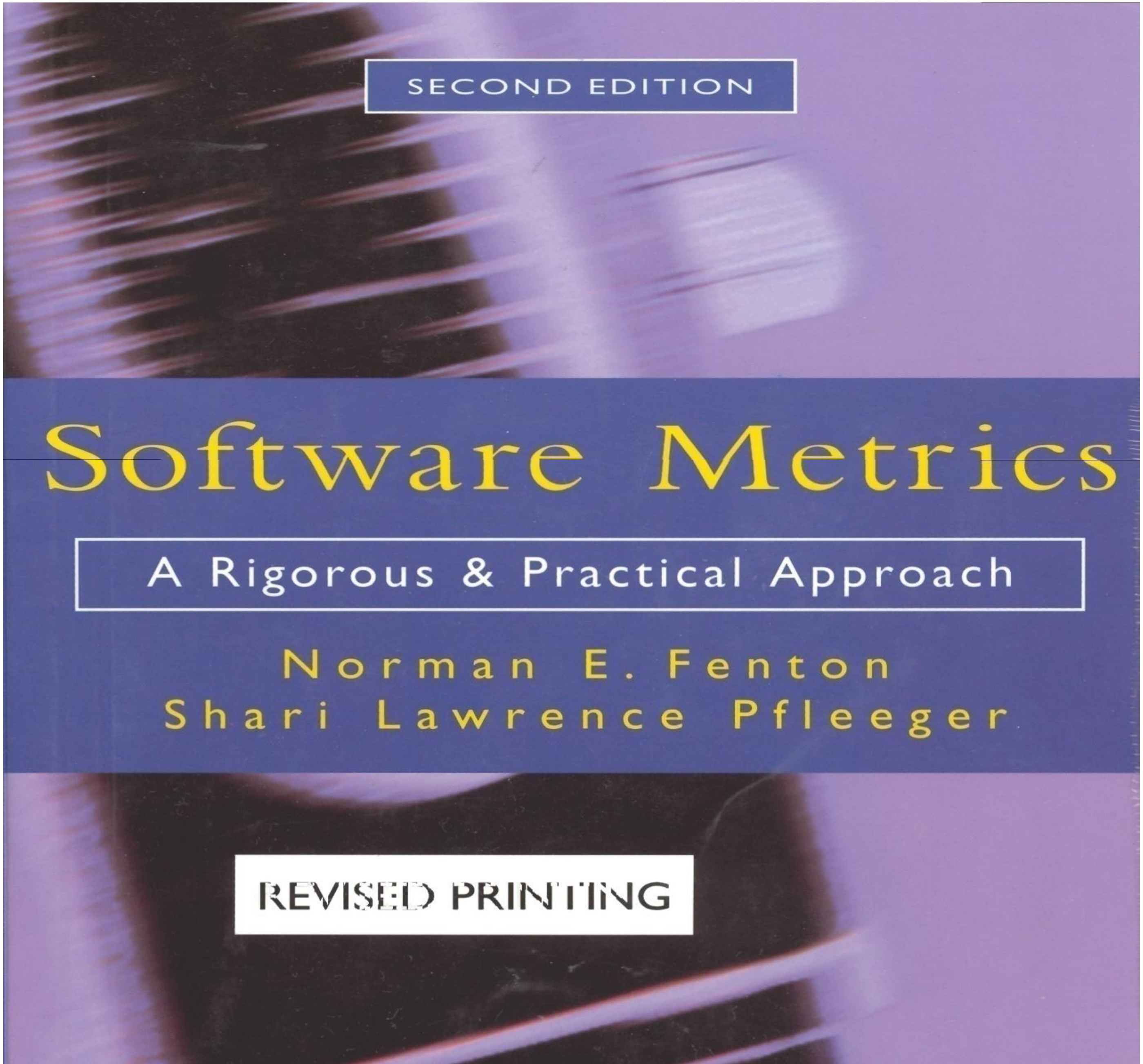

SECOND EDITION
Software Metrics
A Rigorous & Practical Approach
Norman E. Fenton
Shari Lawrence Pfleeger
REVISED PRINTING

SOFTWARE METRICS

REFERENCE BOOK

- Title : ***Metric and models in software quality engineering***
- Author : ***Stephen H.Kin***
- Publisher : ***Addison Wesley 1995 Pearson Education 2003***
- Edition : ***Thomson Delmar 2nd edition.***
- William A. Florac and Areitor D.Carletow,

"Measuring Software Process ", Addison Wesley, 1995.


SECOND EDITION
Software Metrics
A Rigorous & Practical Approach
Norman E. Fenton
Shari Lawrence Pfleeger
REVISED PRINTING

Objective

- Overview of software metrics
- The basics of measurement
- Framework of software measurement
- Empirical Investigation

Content

- What is measurement
- What are the software metrics?
- Scope of software engineering metrics: a chronological review.

Key Points

- **Metric - ** A quantifiable characteristic of software.
- **Measurement - ** The process of mapping from real world attributes to a mathematical representation.
- **Model - ** A mathematical relationship between metrics.

Why Measure?

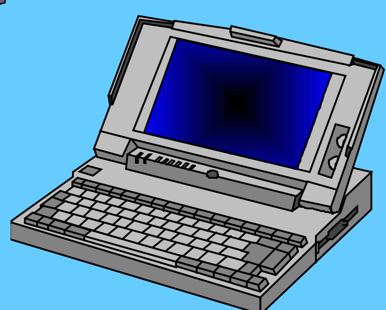
- The primary purpose of a measurement program should be to helping project managers make project decisions.
- You can't manage what you can't measure!
- If you are not measuring, then how are you managing?

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**
- entity:-object(person or room)
- Event- Journey or testing phase of a software project)
- Attribute- property of an entity(color of room , cost of journey)

Measurement Defined

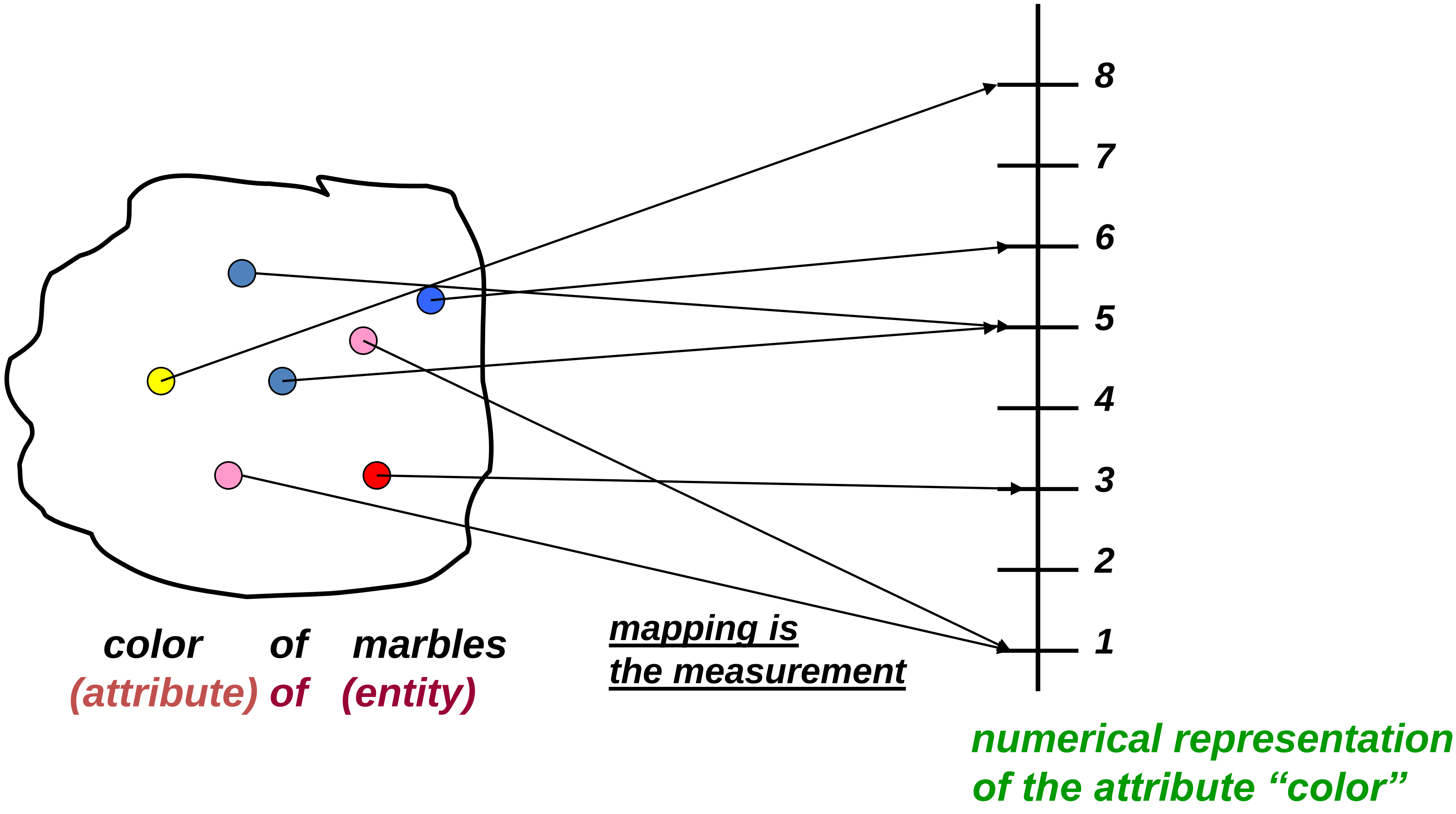
Entity:



Attribute:
Features & Properties

Mapping:
Numbers
&
Symbols

Measurement Example



Motivation for Metrics

- Estimate the cost & schedule of future projects
- Evaluate the productivity impacts of new tools and techniques
- Establish productivity trends over time
- Improve software quality
- Forecast future staffing needs
- Anticipate and reduce future maintenance needs

What is Software Metrics

- Provides values and units for describing software attributes

Why Measure Software?

- Helps in:
  - Comparing multiple approaches
  - Estimating manpower/time required
  - Find complex components in a system
  - Predict quality

Measuring for Understanding, Control and Improvement

- Measure help us for following three activities
  - Understand what happen during development and maintenance.
  - Control what happen on the project , predict what happen and make changes to meet our goal.
  - Based on the design quality Improve the process and product.

Measurement in Everyday Life

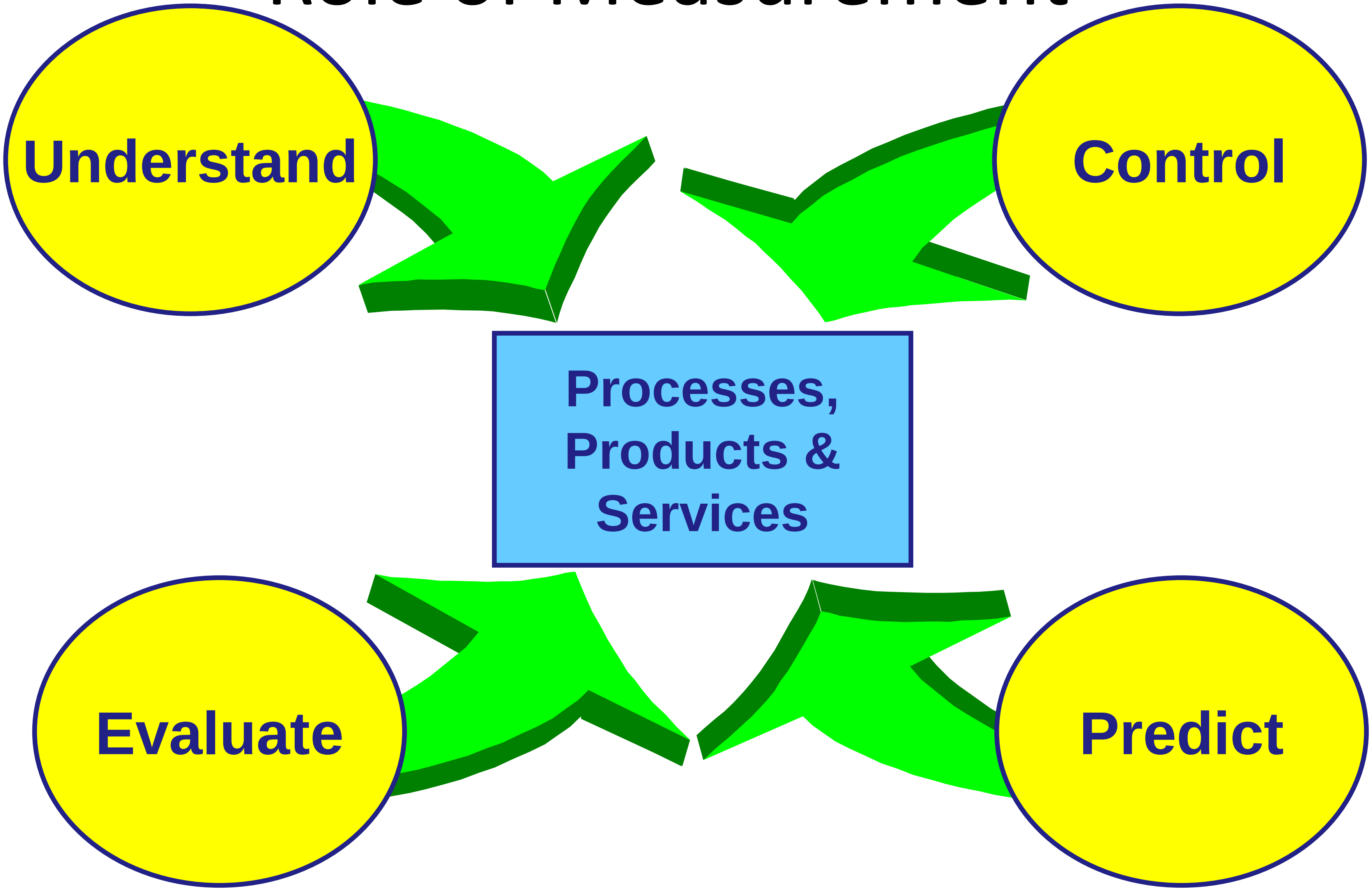
- How do we use measurement in our lives?
  - In a shop, price is a measure of the value of an item, and we calculate the bill to make sure we get the correct change.
  - Height and size measurements ensure clothing will fit correctly.
  - When traveling, we calculate distance, choose a route, measure speed, and predict when we'll arrive
- Measurement helps us to:
  - Understand our world
  - Interact with our surroundings
  - Improve our lives

- Some aspect descriptor allows us to compare it with others(calculation are done with set of rules)
- Price=one item with other
- Distance=distance travel to distance remaining

Measurement in Everyday Life

- Making Things Measurable
  - "What is not measurable, make measurable" (Galileo Galilei)
    - One aim of science is to find ways of measuring attributes of things we're interested in.
    - Measurement makes concepts more visible, therefore more understandable and controllable.
    - Attributes previously thought to be unmeasurable now form basis for decisions affecting our lives (e.g., air quality, inflation index).
  - Measuring the unmeasurable improves understanding of particular entities, attributes
    - Act of proposing a particular measure can open discussion that will lead to greater understanding
    - Making new measurement may requiring modifying environment or practices (e.g., using a new tool, adding a step in a process)

Role of Measurement



Measurement in Everyday Life

- It helps us to understand our world,interact with our surroundings and improve our lives.
  - Economic indicators determine prices, pay raises
  - Medical system-diagnose specific illness
  - Atmosphere-weather prediction
  - Journey-calculate distance,speed

Making Things Measurable

- Measurement:-direct quantifiation.eg-height of the tree
- Calculation:- indirect, quantification.

  eg-city inspectors assign a valuation to a house. factors-number of rooms , overall floor space.
- Two kinds of quantification:
  - Measurement
  - Calculation

Metric Classification

- Products
  - Explicit results of software development activities
  - Deliverables, documentation, by products

- Processes
  - Activities related to production of software

- Resources
  - Inputs into the software development activities
  - hardware, knowledge, people

Types of Measures

- Direct Measures (internal attributes)
  - Cost, effort, LOC, speed, memory

- Indirect Measures (external attributes)
  - Functionality, quality, complexity, efficiency, reliability, maintainability

Size-Oriented Metrics

- Size of the software produced
- *LOC* - Lines Of Code
- *KLOC* - 1000 Lines Of Code
- *SLOC* – Statement Lines of Code (ignore whitespace)
- Typical Measures:
  - Errors/KLOC, Defects/KLOC, Cost/LOC, Documentation Pages/KLOC

LOC Metrics

- Easy to use
- Easy to compute
- Language & programmer dependent

Software Metrics

- An indirect measure involves a measure and a prediction formula.

- For example, you cannot calculate density directly. First you measure mass and volume, then you can calculate density=Mass/Volume.

Software Metrics

- In computer science, many of the "ilities" cannot be measured directly. These include:

- maintainability
- readability
- testability
- quality (not a true "ility")
- reusability
- complexity (another non-"ility")

Some measures used in SE

Entity	Attributes	Measures
Completed project	Duration	Months from start to finish
Completed project	Duration	Days from start to finish
Program code	Length	Number line of code (LOC)
Program code	Length	Number of executable statements
Integration testing process	Duration	Hours from start to finish
Integration testing process	Rate at which faults are found	Number of faults found per KLOC
Tester	Efficiency	Number of faults found per KLOC
Program code	Reliability	Rate of occurrence of failure in CPU time

Direct & Indirect Measurement

- Direct measurement of an attribute of an entity involves no other attribute or entity
  - Length of source code (measured by LOC)
  - Duration of testing process (measured by elapsed time in hours)
  - Number of defects discovered during the testing process (measured by counting defects)
  - Time a programmer spends on a project (measured by weeks works)

Indirect Measurement

Programmer productivity	$\frac{\text{LOC produced}}{\text{Person month of effort}}$
Module defect density	$\frac{\text{Number of defects}}{\text{module size}}$
Defect detection efficiency	$\frac{\text{Number of defect detected}}{\text{total number of defects}}$
Requirement stability	$\frac{\text{number of initial requirement}}{\text{Total number of requirement}}$
Test effectiveness ratio	$\frac{\text{Number of items covered}}{\text{Total number of items}}$
System Spoilage	$\frac{\text{Effort spent fixing faults}}{\text{total project effort}}$

Measurement in Software Engineering

- Why measurement in software engineering.
- SE activities include managing, costing, planning, modeling, analyzing, specifying, designing, implementing, testing and maintaining.
- Engineering approach-each activity is understood and controlled.
- CS provides the theoretical foundation for building software.
- SE focuses on implementing the software in a controlled and scientific way.

Importance of SE

- Used in
  - oven,
  - banking transaction
  - Air traffic control
  - Sophisticated power plants
  - Weapons
  - Quality of life depends on software

  Worse-too many stories about software that put lives and businesses at risk.

Software Metrics

- The following are criteria for valid metrics:

1. A metric must allow different entities to be distinguished.

2. A metric must obey a representation condition.

A **representation condition** is the condition that the rules assigning the metric properly characterizes the attribute measured.

Software Metrics

3. Each unit of the attribute must contribute an equivalent amount to the metric.

For example, if we are counting lines of code, the attribute would be "lines of code" and we do not give different weights to different lines of code. Each line counts "one".

Software Metrics

4. Different entities can have the same attribute values.

For example, horses and people both have the attribute “height” which can be “measured” using a “metric”.

Software Metrics

- Many times the attribute of interest is not directly measurable.

- In this case an indirect measure is used.

- **For example, you can’t directly measure the temperature on the surface of the sun, so an indirect measure is used.**

Measurement in other Engineering Discipline

- Eg-Designing electrical circuits-theories ohm law, describes the relationship between resistance, current and voltage in the circuit.
- Measuring the changes in behavior, measuring the cause and effect.
- Build the circuit with specific current and resistance ,we know voltage is required
- Its difficult to imagine electrical, mechanical and civil engineering without measurement

Measurement in SE

- Before measurement project can be planned
  - Objective and scope should be established
  - Alternative solution should be considered
  - Technical and management constraints should be identified
- This information is required to estimate costs, project tasks and project schedule.

Objectives for Software Measurement

- Measurement is not only useful but necessary.
- How can you tell if your project is healthy if you have no measures of its health?
- So measurement is needed at least or assessing the status of your projects, products, process and resources.
- Measurement action is motivated by a particular goal.

Kind of Information need to Manager

- What does each process cost?
  - Eliciting requirements, specifying,designing,coding and testing.

Productivity of staff

- How productive is the staff?
  - Manger figure out, how productive the staff is at each activity.

- How good is the code being developed?
  - Measure software quality, compare different products, predict the effect of changes.

- Will the user be satisfied with the product?
  - Requirement requested have actually been implemented properly

- How can be improve?
  - Compare two design method to see which one yields the higher quality code.

Engineers

- Are the requirement testable?
  - Analyze each requirement –reliable
- Have we found all faults?
  - Measure the number of faults and find the root causes
- Have we met our product or process goal?
  - Measure characteritics of products and process-met standards(certification, eg-no module contain more than 100 lines of code)
- What will happen in the future?
  - Measure the existing and current product and predict the future.

The Scope of Software Metrics

- Cost and effort estimation
- Productivity measures
- Data collection
- Quality models and measures
- Reliability models
- Performance evaluation and models
- Structural and complexity metrics
- Capability-maturity assessment
- Management by metrics
- Evaluation of methods and tools

1.Cost and Effort Estimation

- Manager role
  - Predict the project cost and time in planning stage.
  - Cost estimation models like
    - COCOMO.
    - Putnam’s.
    - SLIM model.

    Effort is measured by

    - Size of the product, capability of developers , level of reuse

    Size id measured by

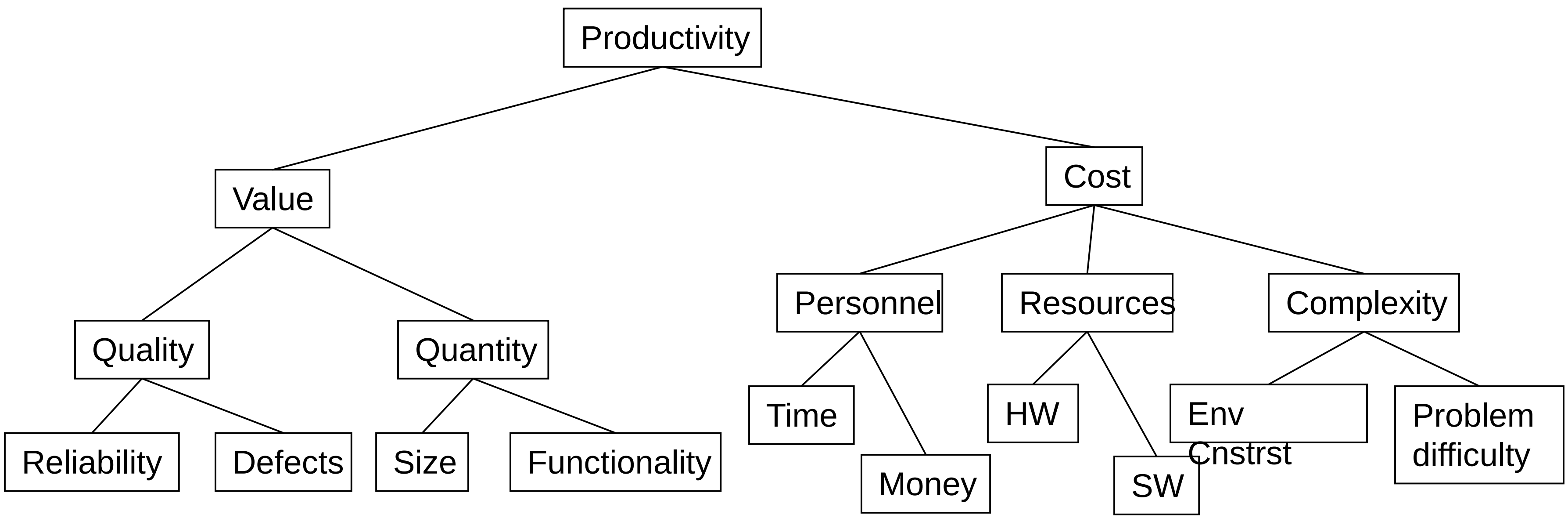
    - Lines of code or Function point

2.Productivity Models and Measures

- The rate of output per unit of input
- Productivity=size/effort
- Productivity=LOC/person-month
- Productivity--------→Value

  Cost

Productivity



3.Data Collection

- Quality of measurement program dependent on data collection.
- Data collection must be planned and executed in a careful and sensitive manner.
- Its essential for scientific investigation of relationship and trends.
- Eg-software failure data collection
  - Time of failure
  - Time interval between failures
  - Cumulative failure up to a given time.
  - Failure experienced in a time interval

4.Quality Models and Measures

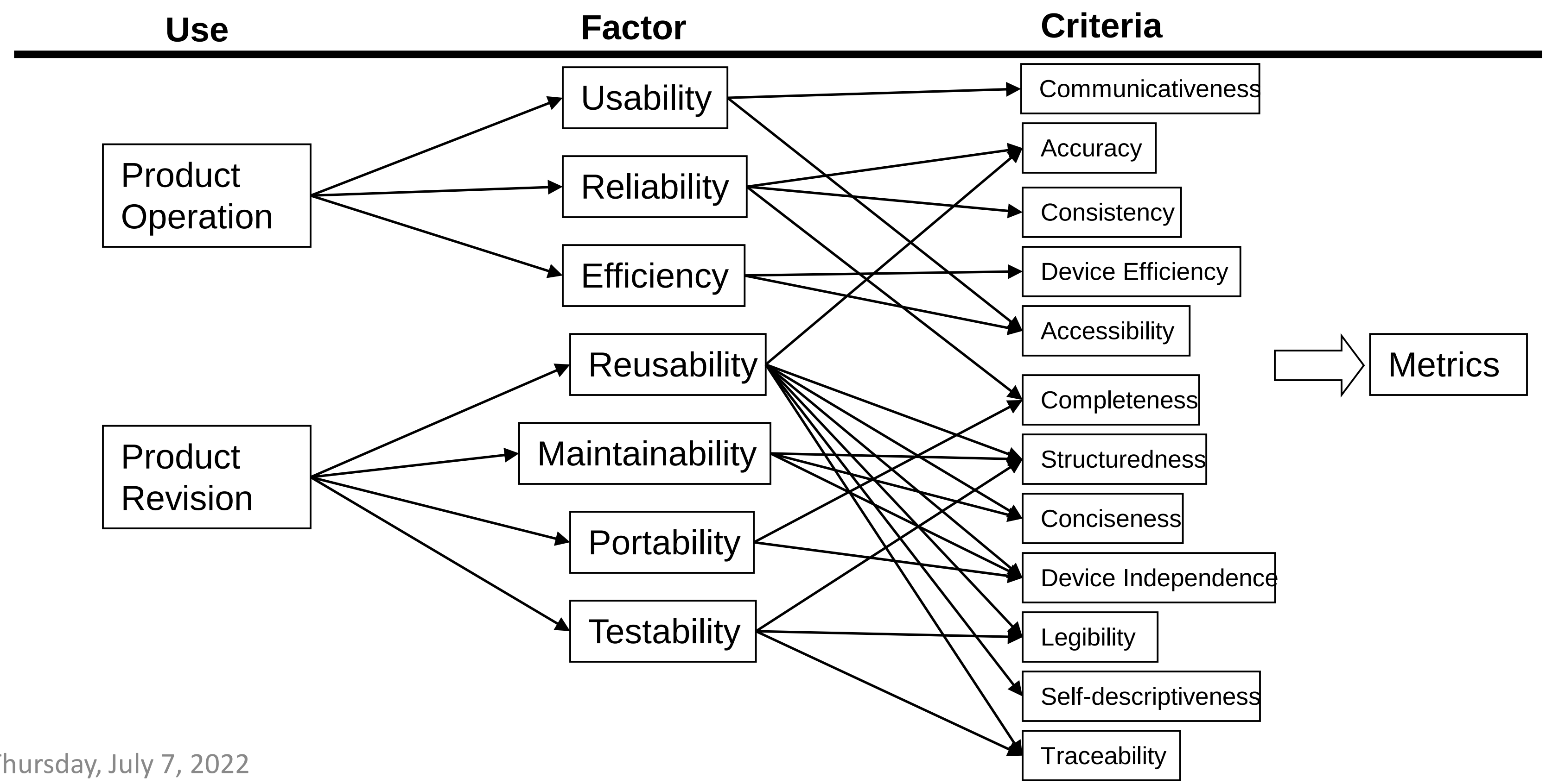
- Assessment of product quality, speed of production is meaningless.
- Quality combined with productivity.
- Measure the factor in terms of dependent criteria
  - Eg-shirt material

Quality Models and Measures

Upper branches-high-level quality-quantify

Lower-level –structuredness and traceability

Quality factor



5.Reliability Models

- Most quality models include reliability as a component factor.
- The most famous model are basic exponential model and logarithmic poisson model.
  - Exponential model assumes Finite failure in infinite time
  - Logarithmic Poisson Model-infinite failure

6.Performance Evaluation and Models

- Performance is another aspect of quality.
- Response time and completion rates.
- Efficiency of algorithm

7.Structural and Complexity Metrics

- Quality like reliability and maintainability cannot be measured until code is available.
- We are able to predict which part of software system less reliable , more difficult to test , even before the system is complete.
- In advance we try to establish, predictive theories to support quality assurance , quality control , quality prediction.

Structural and complexity Metrics

- Complexity metric
  - Cyclomatic complexity(McCabe 1989)-defining the number of independent path in execution of a program.

8.Management by Metrics

- Measurement is important in SPM.
- Customer and developer rely on, measurement based charts and graphs to decide If the project is on track.
- Project can be compared and contrasted ,by standards of measurement and testing report.
- Customer is not well-versed In software terminology, so measurement can paint a picture of progress in general,understandable terms

9.Evaluation of Methods and Tools

- Many article and book describe new methods and tools , that make more productive and your products better and cheaper.
- Many org perform experiment ,run case studies or administrative survey to help them decide whether a method a tool is make a positive difference in particular situation.
- Success depends on good experiemental

10.Capability Maturity Assessment

- US Software Engineering Institute (SEI) model (1989):CMM grading using five-level scale for software development and maintenance.
- Level 1-Initial
- Level 2-Repeatable
- Level 3-Defined
- Level 4-Managed
- Level 5- optimizing

Determining what to measure

- The SEI has suggested that there are five levels of process ranging from ad hoc, repeatable, defined, managed and optimizing.
- The SEI distinguishes one level from another in terms of key process activities going on at each level.

Overview of process maturity and measurement

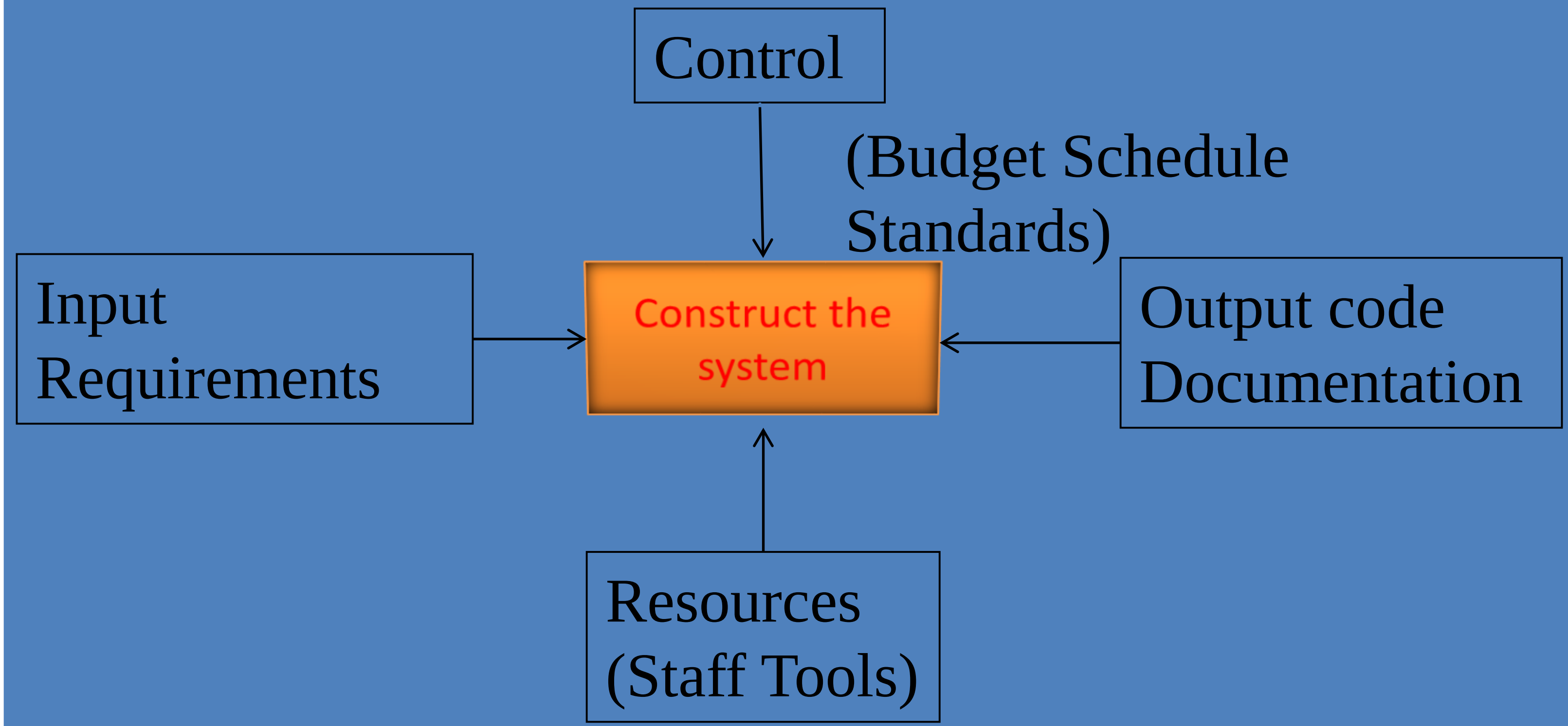
Maturity level	Characteristics	Type of metrics to use
5. Optimizing	Improvement feedback to the process	Process plus feedback for changing the process
4. Managed	Measured process	Process plus feedback for control
3. Defined	Process defined and institutionalized	Product
2. Repeatable	Process dependent on individuals	Project management
1. Initial	Ad hoc	Baseline

Level 1 : Initial

- Inputs to the process are ill-defined; while outputs are expected, the transitions from inputs to outputs is undefined and uncontrolled.
- Similar projects may vary widely in their productivity and quality characteristics because of lack of adequate structure and control.
- Baseline measurements are needed to provide starting point for measuring improvement as maturity increases.

Level 2: Repeatable

- It identifies the inputs and outputs of the process, the constraints and the resources used in the final product.
- The process is repeatable: proper inputs, proper outputs, but there is no visibility into how the outputs are produced

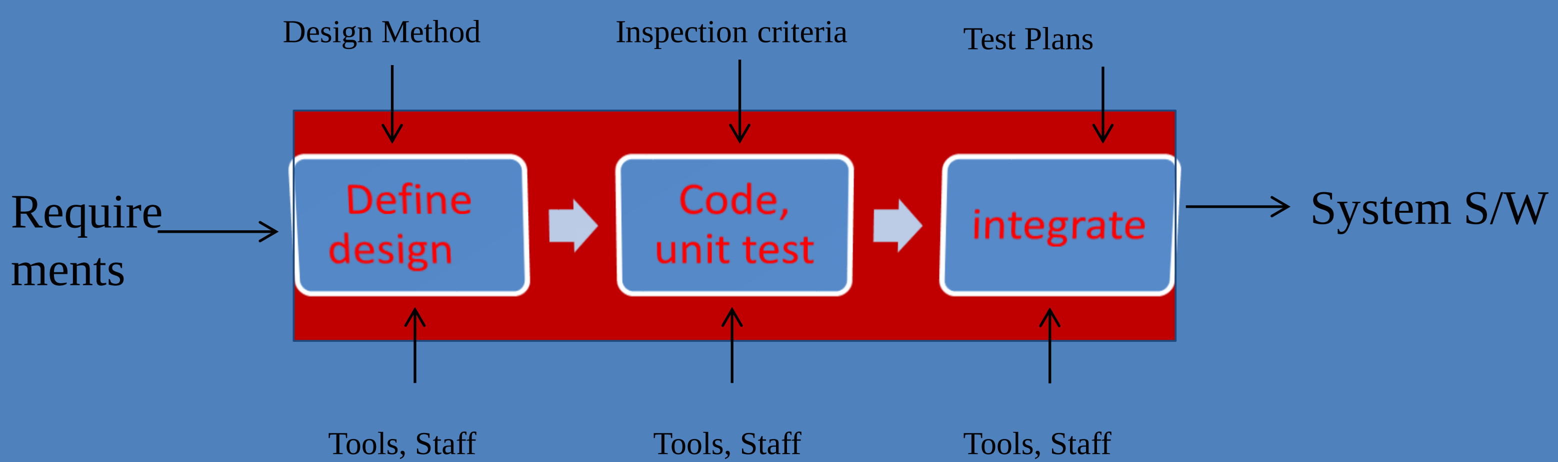


A Repeatable process

Level 3 : Defined

- The defined process provides visibility into the “construct the system” box
- In this level the intermediate activities are defined and their inputs and outputs are known and understood.
- All process can be examined, measured and assessed.

A defined process



Level 4: Managed

- A managed process adds management oversight to a defined process.

- You can compare contrast, the effects of changes in one activity can be traced in the others.

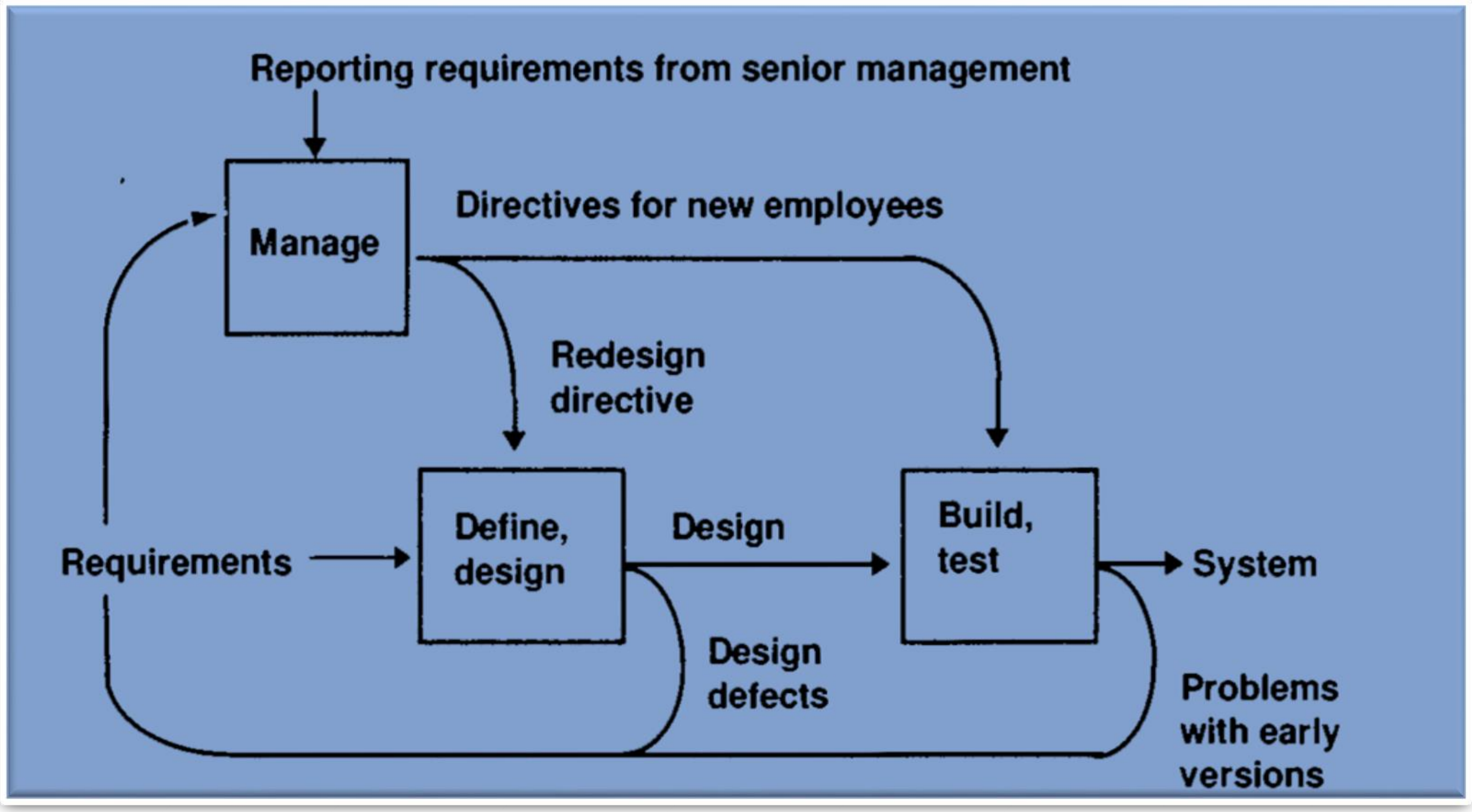
- The feedback determines how resources are deployed.

- You can evaluate the effectiveness of process activities: how effective are reviews? Configuration management? Quality assurance?.

- A significant difference between level 3 and 4 is that level 4 measurements reflects characteristics of the overall process and of the interaction among and across major activities.

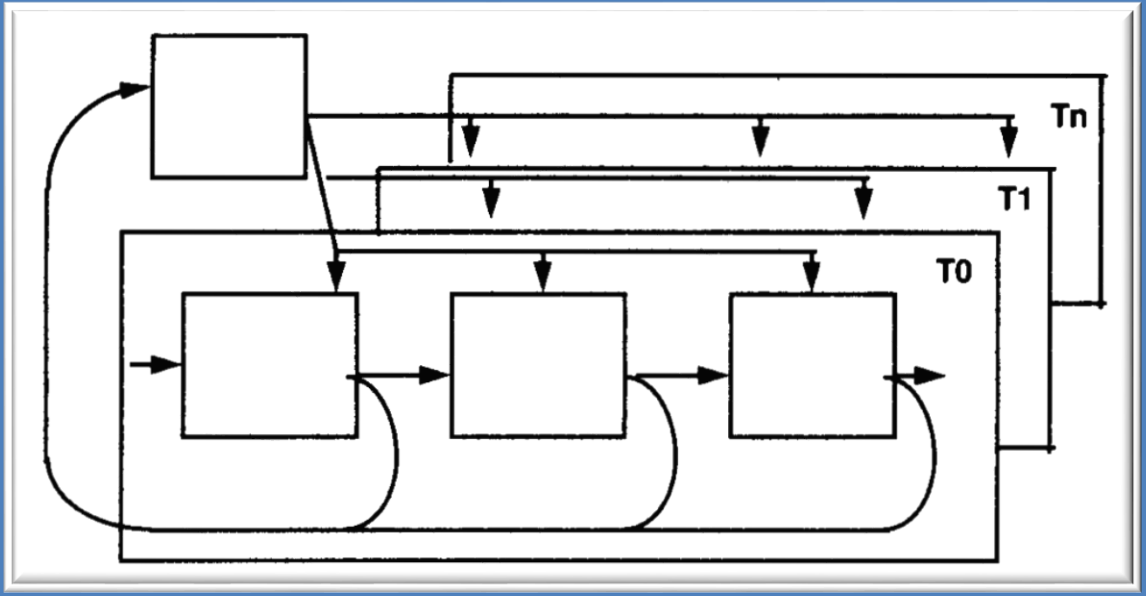
- Management oversight relies on a metrics database that can provide information about such characteristics as distribution of defects, productivity and effectiveness of tasks, allocation of resources, and the likelihood that planned and actual values will match.

- A managed process (Level 4)



Level 5 : Optimizing

- Here measure from activities are used to improve process, possibly by removing and adding process activities, and changing the structure dynamically in response to measurement feedback.



An optimizing process (Level 5)