**Measures**

Measure - quantitative indication of extent, amount, dimension, capacity, or size of some attribute of a product or process. – Number of errors

Metric - quantitative measure of degree to which a system, component or process possesses a given attribute. “A handle or guess about a given attribute.” – Number of errors found per person hours expended

Example Metrics:

Defects rates

Errors rates

Measured by:

– individual

– module

– during development

Errors should be categorized by origin, type, cost

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

**Coupling & Cohesion**

Module: Definition

A logical collection of related program entities

Not necessarily a physical concept, e.g., file, function, class, package

Often requires multiple program entities to express:

– Linked list module may require many class, e.g., list, node, iterators, etc.

Desired Interaction

Minimize external module interaction

– modules can be used independently

– easier to test

– easier to replace

– easier to understand

Maximize internal module interaction

– easier to understand

– easier to test

Characteristics

Cohesion– Internal interaction of the module. Crisp abstraction of purpose

Coupling – External interaction of the module with other modules

Action – Behavior of the module. What it does

Logic – How it performs the behavior. Algorithm used

Context – Specific usage of the module

**Refactoring**

fac·tor – The individual items that combined together form a complete software system:

identifiers

contents of function

contents of classes and place in inheritance hierarchy

fac·tor·ing – Determining the items, at design time, that make up a software system

Refactoring

Process of changing a software system in such a way that it does not alter the external behavior of the code, yet improves its internal structure [Fowler'99]

A program restructuring operation to support the design, evolution, and reuse of object oriented frameworks that preserve the behavioral aspects of the program [Opdyke'92]

**Configuration Management**

Software Configuration Management involves:

Revision control (version control) and change management

Build and release management (diff and patch)

Process and environment management

Facilitate teamwork and collaboration

Issue/Defect tracking