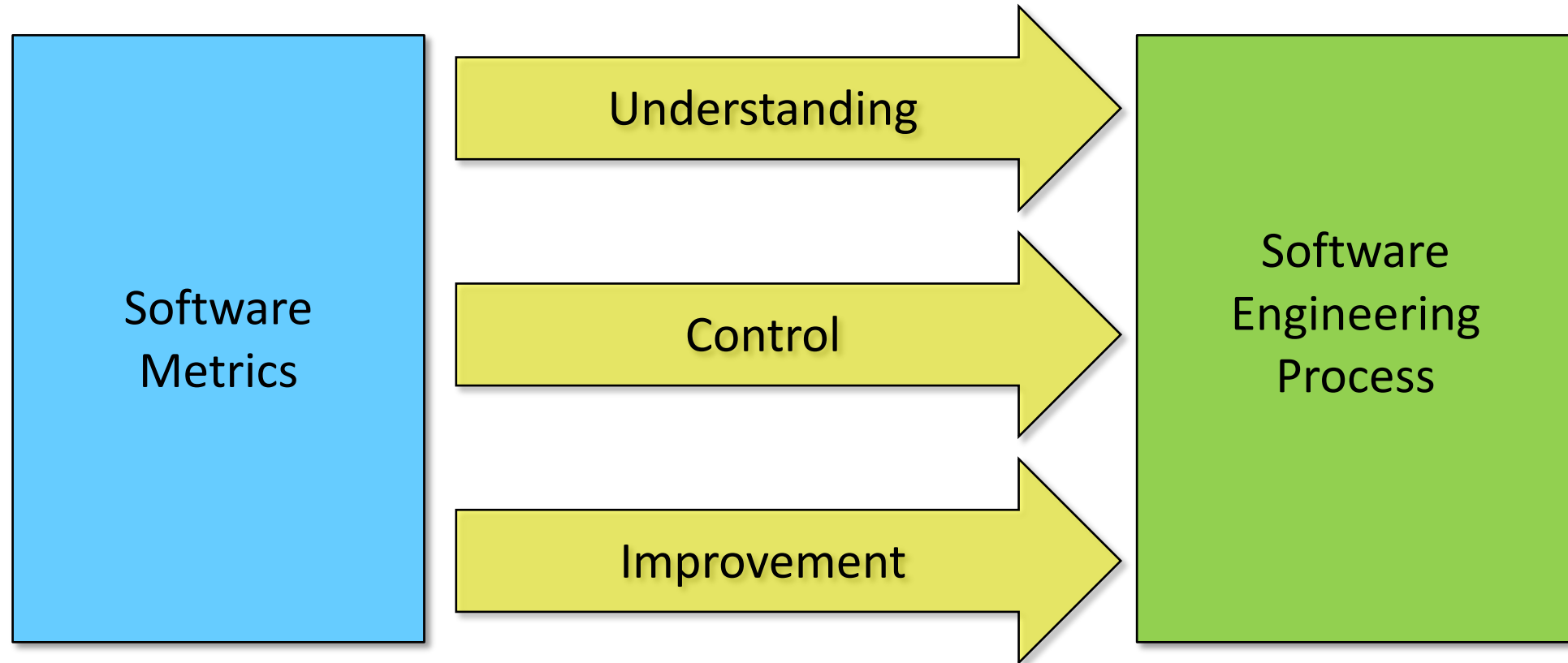


# Software Measurement & Metrics



**SWEN-261**

**Introduction to Software  
Engineering**

Department of Software Engineering  
Rochester Institute of Technology

# What is Software Measurement?

- Measurement is fundamental to any engineering discipline and software engineering is no exception
- Why Measure Software?
  - *Establish the quality of the current product or process.*
  - *To predict future qualities of the product or process.*
  - *To improve the quality of a product or process.*
  - *To determine the state of the project in relation to budget and schedule.*
- Software metrics offer an assessment of the impact of decisions made during software development projects.

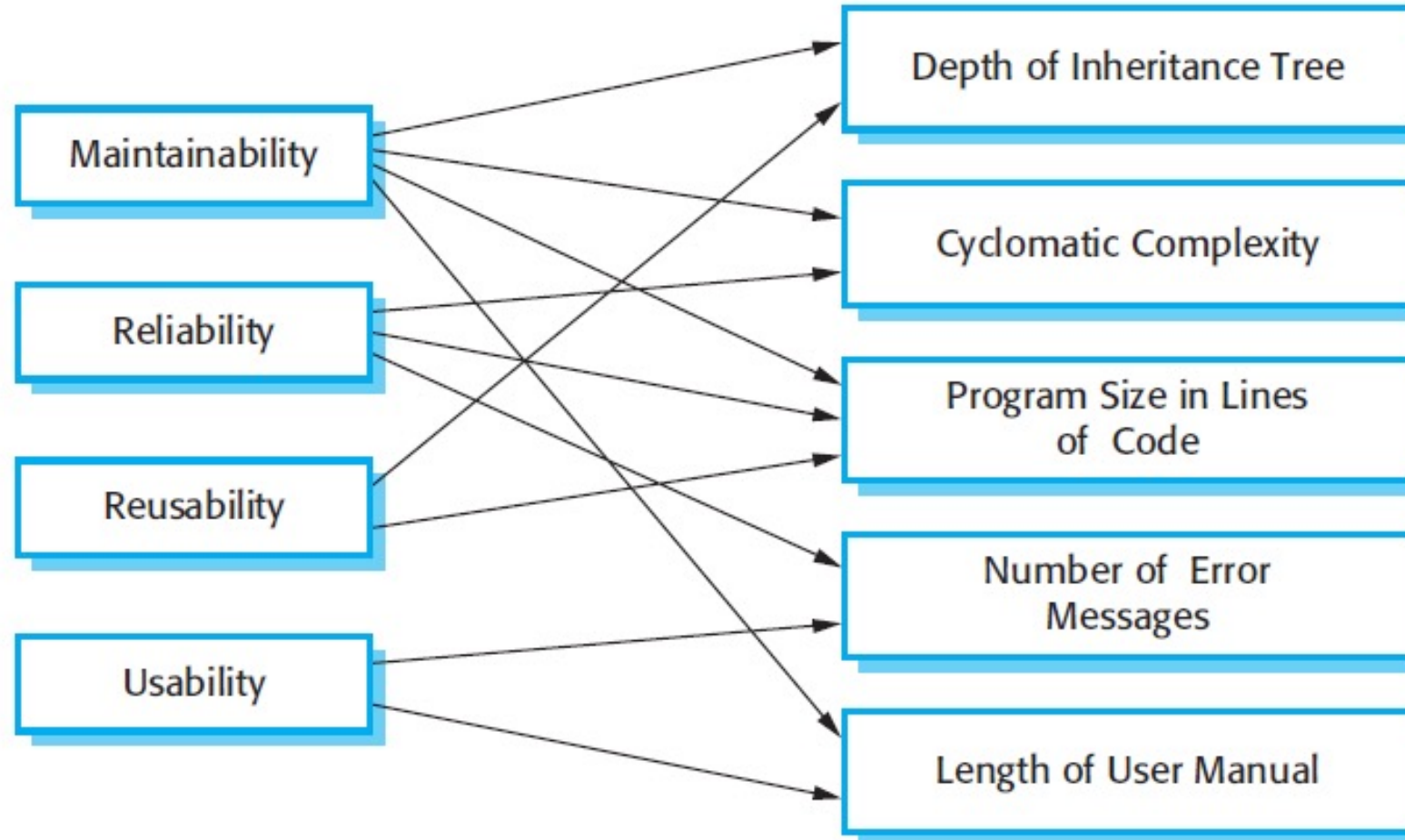
# A metric is not just a number.

- A metric is a quantitative function that calculates some characteristic and produces a numeric measurement which will be used to make a decision.
- For software product development, metrics fall into three broad categories:
  - 1. *Process Metrics***
    - ◆ Activities related to production of software like effort required in the process, time to produce the product, number of defects found during testing, effectiveness of defect removal during development, maturity of the process
  - 2. *Project Metrics***
    - ◆ Describe the project characteristics and execution like number of software developers, cost and schedule, productivity
  - 3. *Product Metrics***
    - ◆ Describe the characteristics of the product, such as size, complexity, design features, performance, efficiency, reliability, portability etc.

# Several software quality attributes can be related to internal attributes through metrics

Software Quality Attributes (external)

Internal Attributes



# Dynamic and Static metrics

- Dynamic Metrics are closely related to software quality attributes
  - *Collected by measurements made of a program in execution*
  - *It is relatively easy to measure the response time of*
  - *They help to assess the efficiency and reliability of a program*
  - *Example: number of bug reports or the time taken to complete a computation*
- Static Metrics have an indirect relationship with quality attributes
  - *Collected by measurements made of representations of the system, such as the design, program, or documentation.*
  - *Static metrics help assess the complexity, understandability, and maintainability of a software system*
  - *Example: code size and the average length of identifiers used*

# Examples of Static Metrics

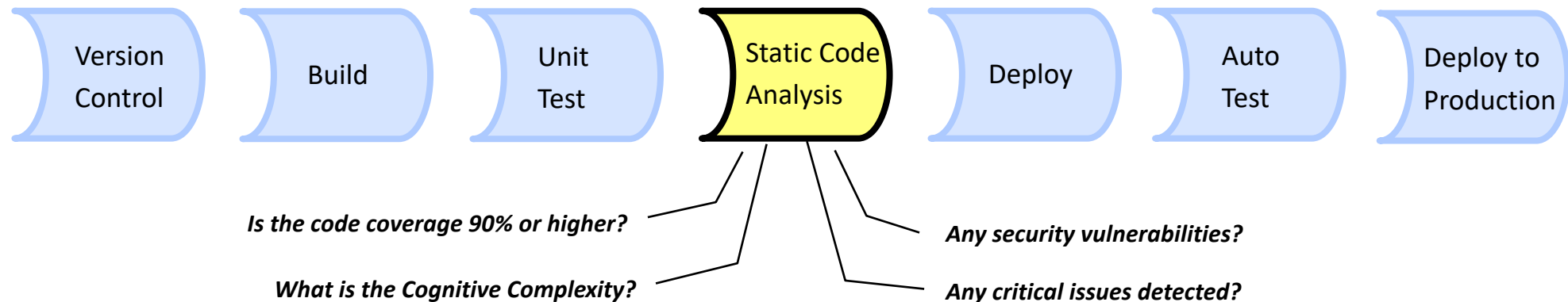
Software Metric	Description
Fan-in/Fan-out	Fan-in is a measure of the number of functions or methods that call some other function or method (say X). Fan-out is the number of functions that are called by function X.
Length of code	Measures the size of a program. Generally, the larger the size of the code of a component, the more complex and error-prone is it likely to be.
Cyclomatic complexity	Used to indicate the complexity of a program. It is a quantitative measure of the number of linearly independent paths through a program's source code.
Length of identifiers	The measure of the average length of identifiers (e.g. names for variables, classes, methods, etc.) in a program. The longer the identifiers, the more likely they are to be meaningful and hence more understandable the program.
Depth of conditional nesting	The measure of the depth of nesting of if-statements in a program. Deeply nested if-statements are hard to understand and potentially error prone.
Code Duplication	Indicates the amount of source code that occurs more than once in a program. Code Duplication is undesirable because it is associated with higher maintenance costs and can be indicative of bad design.
Fog index	Refers to a readability test that aims to determine the level of text difficulty, or how easy a text is to read. The higher the level, the more difficult the document is to understand

# Even though you can count something, it does not necessarily count for anything.

- A metric is only as good as the decisions that it will be used to make.
  - *Measuring something without it having a solid connection to possibly improving what you are doing is a waste of time and resources.*
- A measurement falling outside of a target range is not an absolute indictment.
- Target values for measurements should be set based on a record of past measurements and resulting performance.
  - *Measurement not in some range → some project quality was poorer*
- If measurements that do not fall in the target range indicate a place for additional scrutiny.
  - *For product metrics, they indicate possible "code smells".*
  - *Places to consider for refactoring, redesign, or reimplementation*

# Static Code Analysis Tools

- Static code analysis tools scan all code in a project and seek out vulnerabilities, validates code against industry best practices, and some software tools validate against company-specific project specifications
- They are used by software development and quality assurance teams to ensure the quality and security of code, and project requirements are met.
- They can run within an IDE or integrate through a CI/CD Pipeline





# Static Code Analysis Tools

## ■ Advantages

- *It can find weaknesses in the code at the exact location.*
- *It allows a quicker turn around for fixes.*
- *They can scan the entire code base.*
- *It permits weaknesses to be found earlier in the development life cycle, reducing the cost to fix.*

## ■ Disadvantages

- *They can produce false positives and false negatives.*
- *They can provide a false sense of security that everything is being addressed.*
- *They are only as good as the rules they are using to scan with.*

## ■ Bottom-line

- *These tools can be a cost-effective approach to measure and track software quality metrics but don't replace other key processes like code reviews*

# Static Code Analysis Tools – SonarQube & SonarLint

- **SonarQube** is an open-source platform for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities on 20+ programming languages including Java and TypeScript
  - *It offers reports on duplicated code, coding standards, unit tests, code coverage, code complexity, comments, bugs, and security vulnerabilities*
- **SonarLint** is an IDE extension that identifies and helps you fix quality and security issues as you code.
- Both SonarLint and SonarQube rely on the same static source code analyzers using SonarSource technology

