INTRODUCTION TO SOFTWARE ENGINEERING

Dr.Lê Hùng Tiến

VLU

Introduction To Software Engineering

Carnegie Mellon University
The Practical Software Engineering Series

Software Configuration Management - 1

Course Objective

- Upon completion of this course, students will have the ability to:
  - Understand the discipline and principles of Software Engineering.
  - Understand the evolution of software in industry and the global competitive trends.
  - Understand software process, product and services.
  - Understand software modeling & techniques.
  - Demonstrate an appreciation for the breadth of software engineering.

Lecture Learning Objectives

- Upon completion of this lecture, students will be able to:
  - Understand Software Configuration Management.
  - Understand changes and the discipline of change management.
- Outcomes:
  - Demonstrate knowledge of software configuration management.
  - Be able to manage changes in a software project.
  - Be able to plan, identify, control and audit software projects to ensure that changes are managed accordingly.

The Need To Control Changes - 1

- Have you experienced the following:
  - The latest version of source code disappears?
  - A major bug that was fixed reappears?
  - Wrong version of the code was tested?
  - A fully developed and tested feature is mysteriously missing?
  - A fully tested program suddenly does not work?
  - No one knows which modules were delivered to customers?
  - Undocumented features suddenly appear?

The Need To Control Changes - 2

- Change occurs in every software project. If not managed, it can create confusion and frustration among software engineers.
  - When several people work on the same project, the last one to make changes can destroy the other’s work.
  - When several people work on the same project, an error corrected by one person may impact others due to someone not being notified.
  - When a function is modified, all affected functions in a project will be impacted.
  - In large programs, developed with incremental releases and many versions, every person must know which version to work on and which version of the code should be tested. Without control there will be conflict. No one will know which version is released to customers.

- The method of coordinating and controlling all changes to the software being built is called *software configuration management*.

Definition

Software Configuration Management is:

- A management process for establishing and maintaining consistency of a product’s performance, functional and physical attributes with its requirements, and the design and operational information throughout its lifecycle.

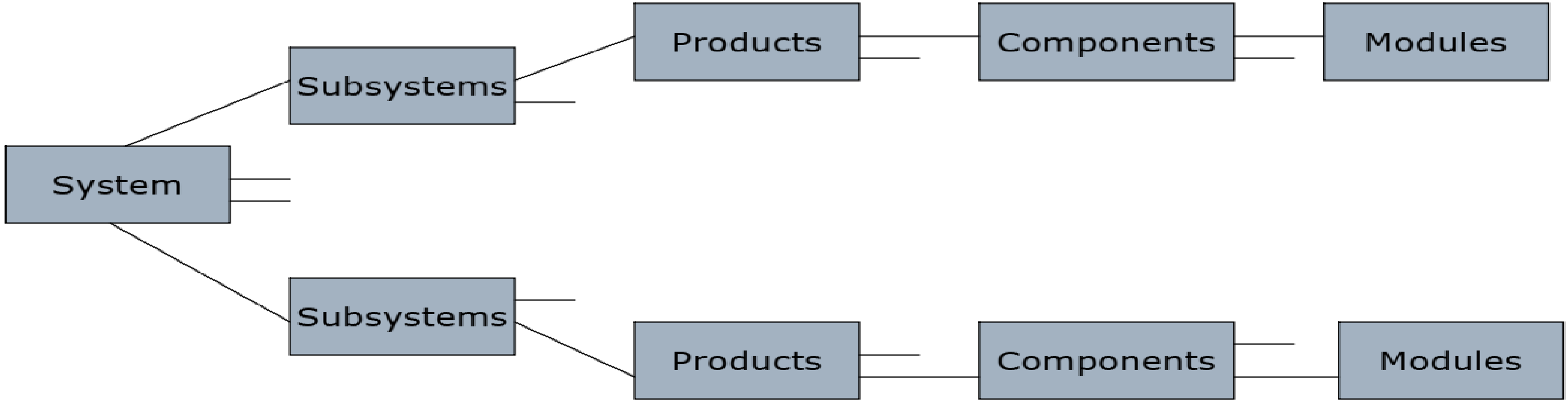
- A discipline applying technical and administrative direction and surveillance to:
  - Identify and document the functional and physical characteristics of a configuration item.
  - Control changes to those characteristics.
  - Record and report change processing and implementation status.
  - Verify compliance with specified requirements.

System Decomposition Terminology

- **System**: The complete package of all the software that meets stakeholder requirements.
- **Subsystem**: A part of a larger system... may have many subsystems, for example:
  - Display, storage, operating system, etc.
- **Product**: Subsystems typically contain many products, for example:
  - Operating systems have a control program, compilers, utilities, etc.
- **Components**: Products may have several components, for example:
  - Control programs have a scheduler, I/O control unit.
- **Modules**: At the lowest level, components consist of many modules which are self contained programs that perform a particular function.

Configuration Items Definition

- Configuration Identification is the process of designating the hardware, software or component elements in a system and recording their characteristics.
- Configuration Identification is the basis from which the configuration of products are defined, verified, and all changes are managed.



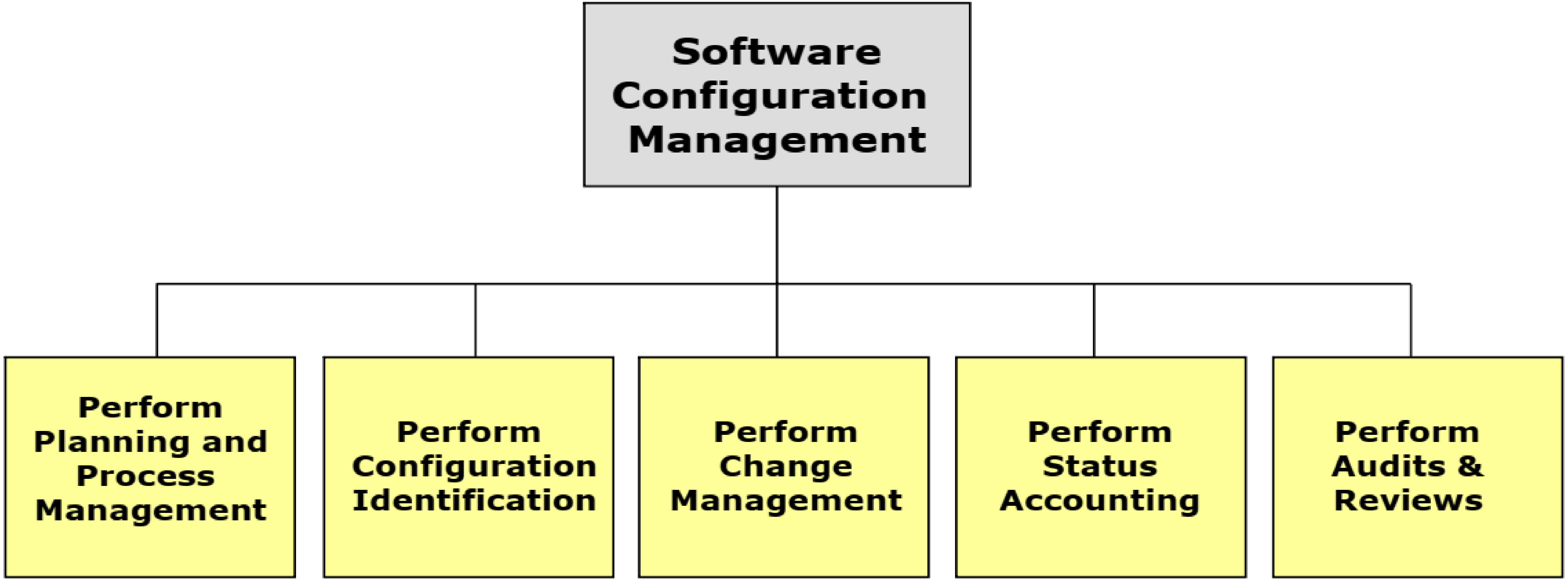
Some Questions …

- □ What is my software configuration?
- □ What is its status?
- □ How do I control change to my configuration?
- □ How do I inform everyone in the project of my changes?
- □ What changes have been made to my software?”
- □ Does anyone else’s change affect my software?
- □ Which version are we working on?
- □ Which version has been tested?

Some Answers ...

Questions	Answers	Discipline
What SCM activities are being performed?	All SCM activities are documented in the SCM plan.	*Perform SCM Planning*
What is my software configuration?	The software consists of the following configuration XXX in version number YYY.	*Perform Configuration Identification*
How are changes made and managed?	The change process is	*Perform Change Management*
What does the current or past revision look like?	The current configuration is XXX and documented in the YYY report.	*Perform Status Accounting*
How do you know changes are correct and the process is followed?	The SCM process is reviewed for compliance per scheduled reviews and audits as planned in the SCM plan.	*Perform Audits and Reviews*

The Configuration Management Disciplines



Software Configuration Management

Perform Planning and Process Management

Perform Configuration Identification

Perform Change Management

Perform Status Accounting

Perform Audits & Reviews

Software Configuration Management (SCM)

- **Purpose:**
  - To establish and maintain the integrity of the products of the software project throughout the software life cycle.

- **Meaning:**
  - Planning software configuration activities.
  - Identifying configuration items.
  - Controlling changes systematically.
  - Maintaining integrity and traceability of the configuration throughout the software life cycle.
  - Audit change processes and software products.
  - Reporting status.

Benefits Of SCM

- □ Visibility of project development and maintenance
- □ Traceability from “as-designed” to “as-built”
- □ Defined and verified configurations
- □ Controlled changes to baselines
- □ Monitored implementation of changes
- □ Recorded and reported configuration of systems
- □ Consistent approach to problem resolution
- □ Applied control throughout system life cycle

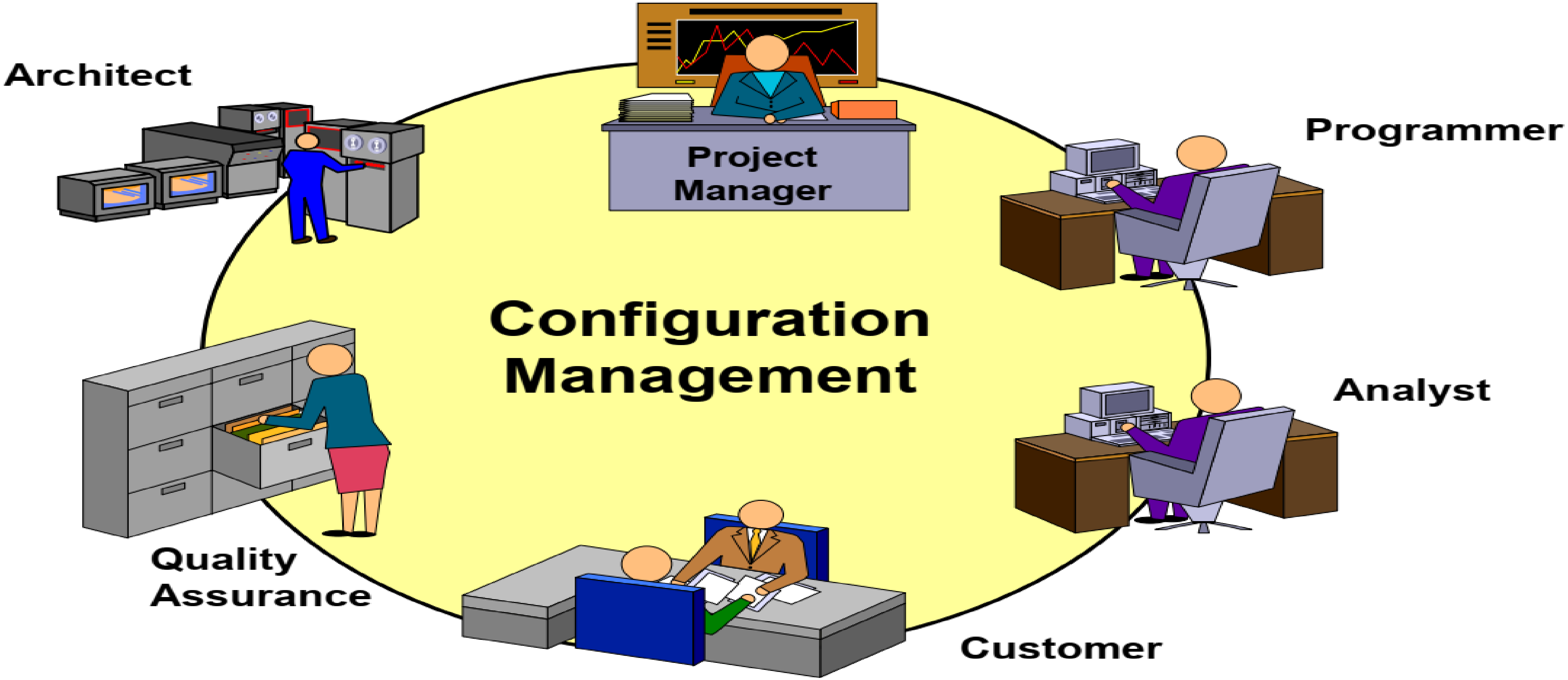
SCM Key Activities

- □ An SCM Plan is prepared for each software project.
- □ The SCM Plan is used as the basis for performing SCM activities.
- □ A Configuration Management library system is established as a repository for the software baseline.
- □ The software products to be placed under configuration management are identified.
- □ Changes to baselines are controlled.
- □ Change requests or problem reports for all configuration items are initiated, recorded, reviewed, approved and tracked.
- □ Products from the software baseline library are created and their release is controlled as per a procedure.
- □ The status of configuration items is recorded.
- □ Measures are collected and used to make business decisions.
- □ The status of configuration items are reported.
- □ Software baseline audits are scheduled and conducted.

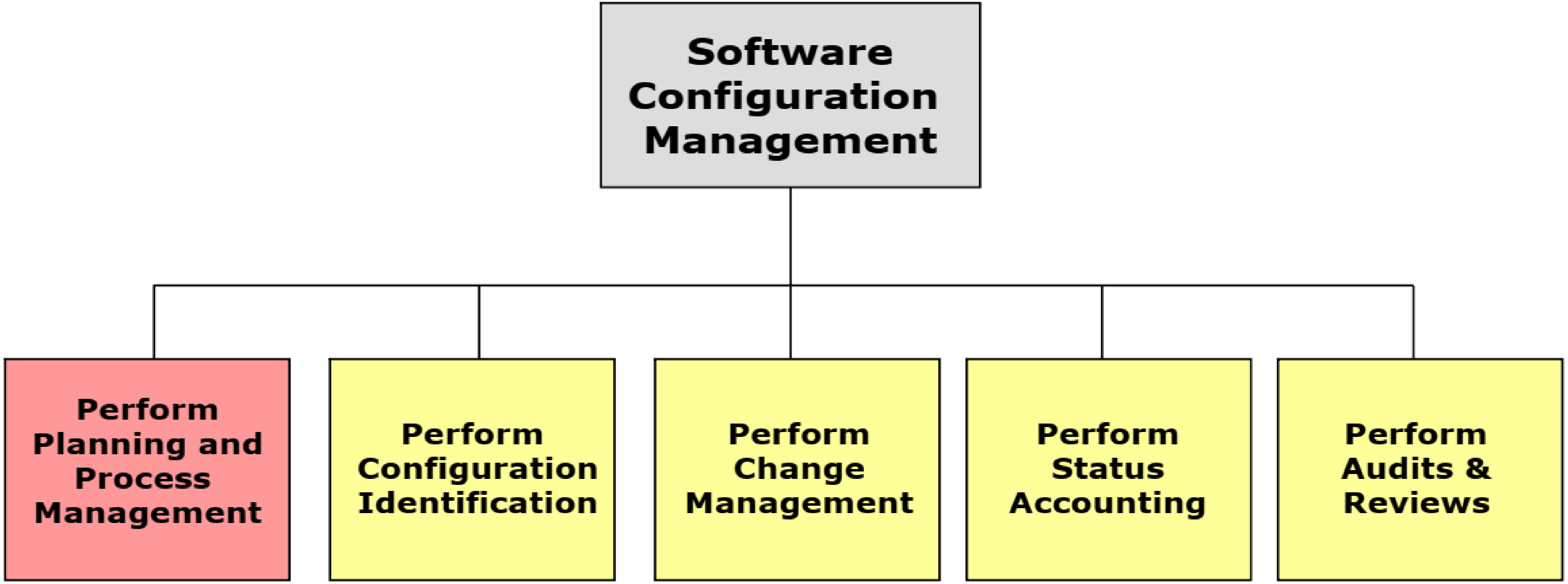
SCM Support Development Environment

- SCM is:
  - A baseline for all requirements, design, code, test, and tools.
  - A control log that documents and describes all changes, and revisions.
  - A controlled environment where changes can be made in an orderly fashion (e.g., check in/out).
  - A protected environment where baseline is kept.
  - A controlled environment where testing can be conducted.

Roles & Responsibilities



The Configuration Management Disciplines


Software Configuration Management
Perform Planning and Process Management
Perform Configuration Identification
Perform Change Management
Perform Status Accounting
Perform Audits & Reviews

SCM Planning - 1

- All SCM activities must be planned early in the project.
- Planning should cover the entire life cycle of the product to ensure an effective, predictable, and repeatable process.
- SCM activities should focus on following the defined process and procedures in order to reduce cost and increase product stability.
- SCM activities must be based on:
  - Operating environment (development / maintenance).
  - Project priorities (schedule, commitments, etc).
  - Customer requirements and support agreements.
  - Existing SCM strengths and weaknesses.

SCM Planning -2

- SCM personnel must conduct training so that all project team members understand their roles and responsibilities and procedures for implementing SCM processes.

- SCM personnel must assess the effectiveness of SCM implementation and performance with metrics (performance indicators).

- Configuration Identification is the basis from which the configuration of products are defined, verified and changes are managed.

Example Of SCM Plan

- Overview
  - SCM goals & objectives
  - Purpose, scope & exceptions
  - Roles & responsibilities
  - SCM process, charter, members, training, tools
  - Product assurance relationship
- SCM Process
  - Configuration item identification
  - Baselines and contents
  - Change management system (including version control)
  - Auditing process
  - Status accounting process
  - SCM support tools (CM tools)
- SCM Practices
  - Reference to organization standards, procedures, guidelines, manual processes
  - SCM forms and records
  - SCM checklists
- SCM Implementation
  - Schedule, milestones, checkpoints
  - Status, reports
  - Audits, verification results
  - Required resources (people, equipment, services)

SCM Plan Summary

- Creating and documenting the SCM Plan is hard work but critical for every software project.

- SCM personnel must:
  - Document product composition
  - Identify what's important to the project
  - Standardize SCM procedures according to defined processes
  - Satisfy SCM requirements

- An SCM plan is the glue that holds the software development process together.
- SCM plans can reduce frustration and ensure an orderly control of changes within a project.

Planning Checklist

□ Is the SCM plan fully documented, reviewed, and approved?

□ Does the SCM plan identify all configuration items to be controlled?

□ Does the SCM plan identify organization standards, procedures and guidelines to be used?

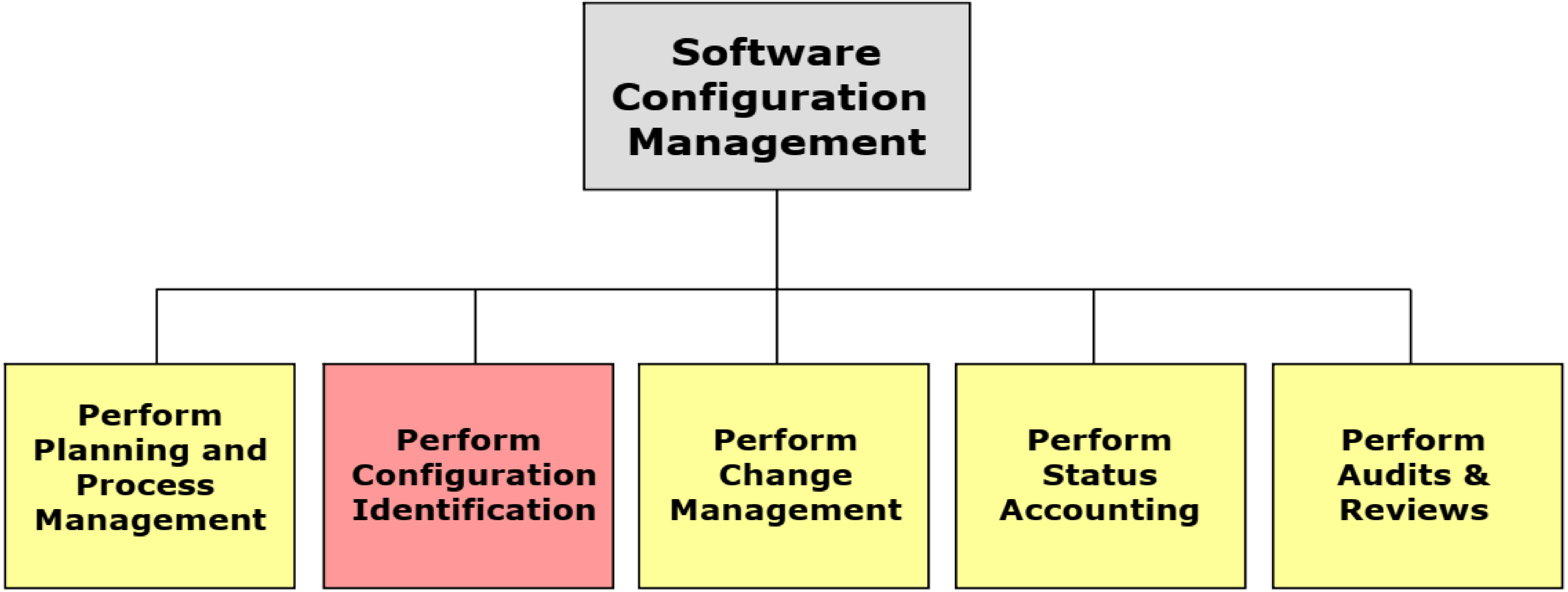
□ Is the SCM process clearly defined and documented in the plan?

□ Does the SCM plan identify required status accounting reports and their frequency?

□ Does the SCM plan identify audits to ensure SCM verification?

□ Does it indicate when and what kind of audits will be conducted?

The Configuration Management Disciplines


Software Configuration Management
Perform Planning and Process Management
Perform Configuration Identification
Perform Change Management
Perform Status Accounting
Perform Audits & Reviews

Configuration Identification

- Configuration Identification is the process of designating the hardware, software or document elements in a system and recording their characteristics.

- Configuration Identification is the basis from which the configuration of products are defined, verified and changes are managed.

- Configuration Items (CIs) are items or groups of items whose external physical and functional characteristics are **defined and controlled** to ensure end-use functions are effectively and efficiently maintained.

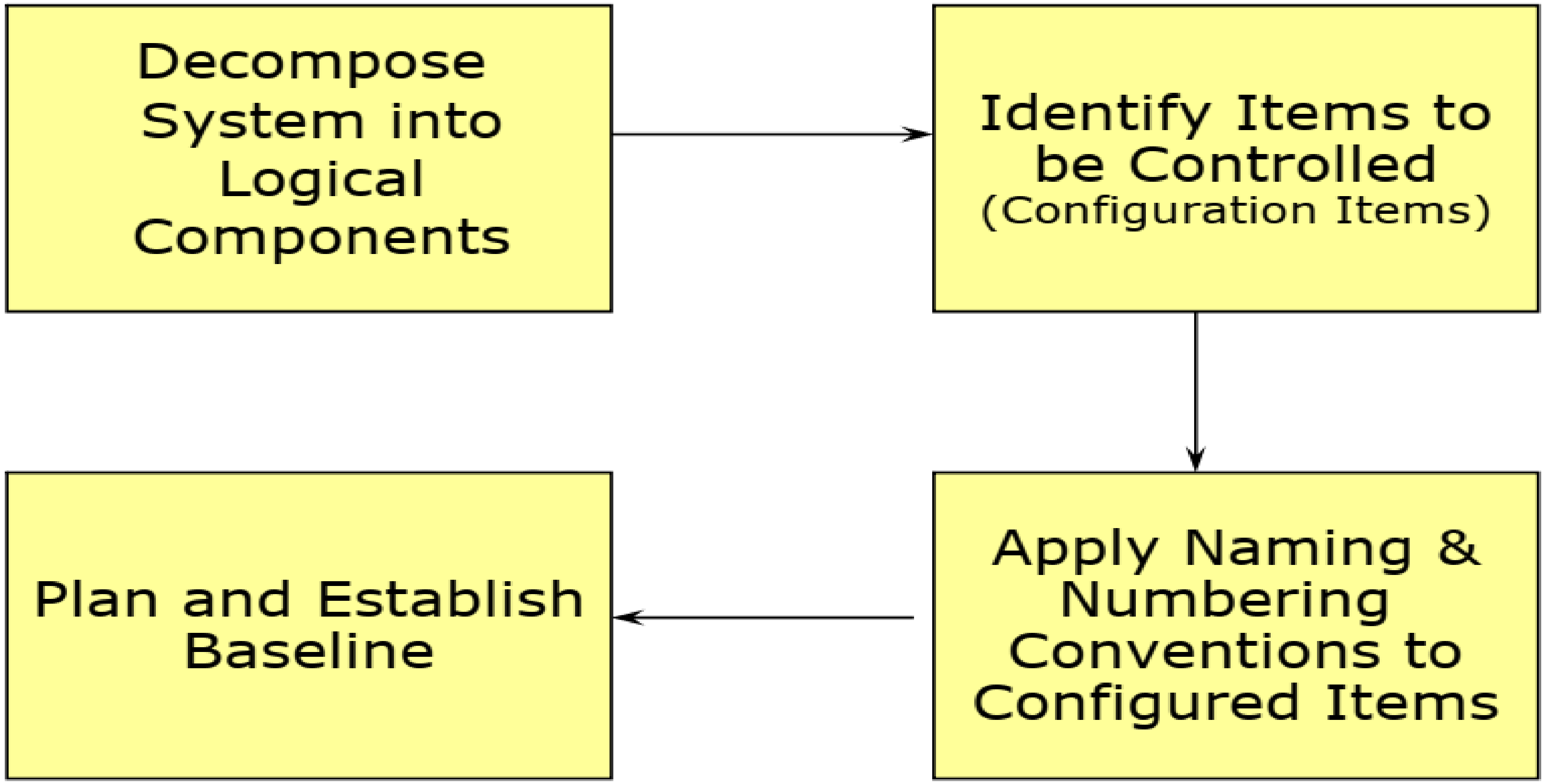
Configuration Items Definition

- The Configuration Identification process starts with the identification of Configuration Items (CIs).
- Configuration Items are identified at various levels of system decomposition, starting with the system, and continuing through the subsystem to the appropriate lowest level of decomposition.
- Configuration Items represent physical entities that are defined, built, tested, delivered and maintained for the customer.
- Configuration Items are units of functional capability that are identifiable and understandable by the system user.
- Configuration Items are the building blocks of system releases.
- Configuration Items consist of a set of components which include documentation, source code, models, test reports, test data, etc.

Why Configuration Items?

- Configuration Items are used to define, build, test, deliver, and maintain units of functional capability that have meaning to the customer.
- Configuration Items are used to allocate and trace functional system requirements.
- Configuration Items are used to facilitate communication between developers and stakeholders.
- Configuration Items are used as the building blocks of planned system releases, for both testing and production.
- Configuration Items are used as a means of planning, estimating, and tracking requested changes to previously delivered systems.
- Configuration Items are used to:
  - Support communication/understanding
  - Support system control
  - Support system maintenance
  - Support reuse of functions
  - Support project cost management

Manage & Control Configuration Items



Decompose System into Logical Components

Identify Items to be Controlled (Configuration Items)

Apply Naming & Numbering Conventions to Configured Items

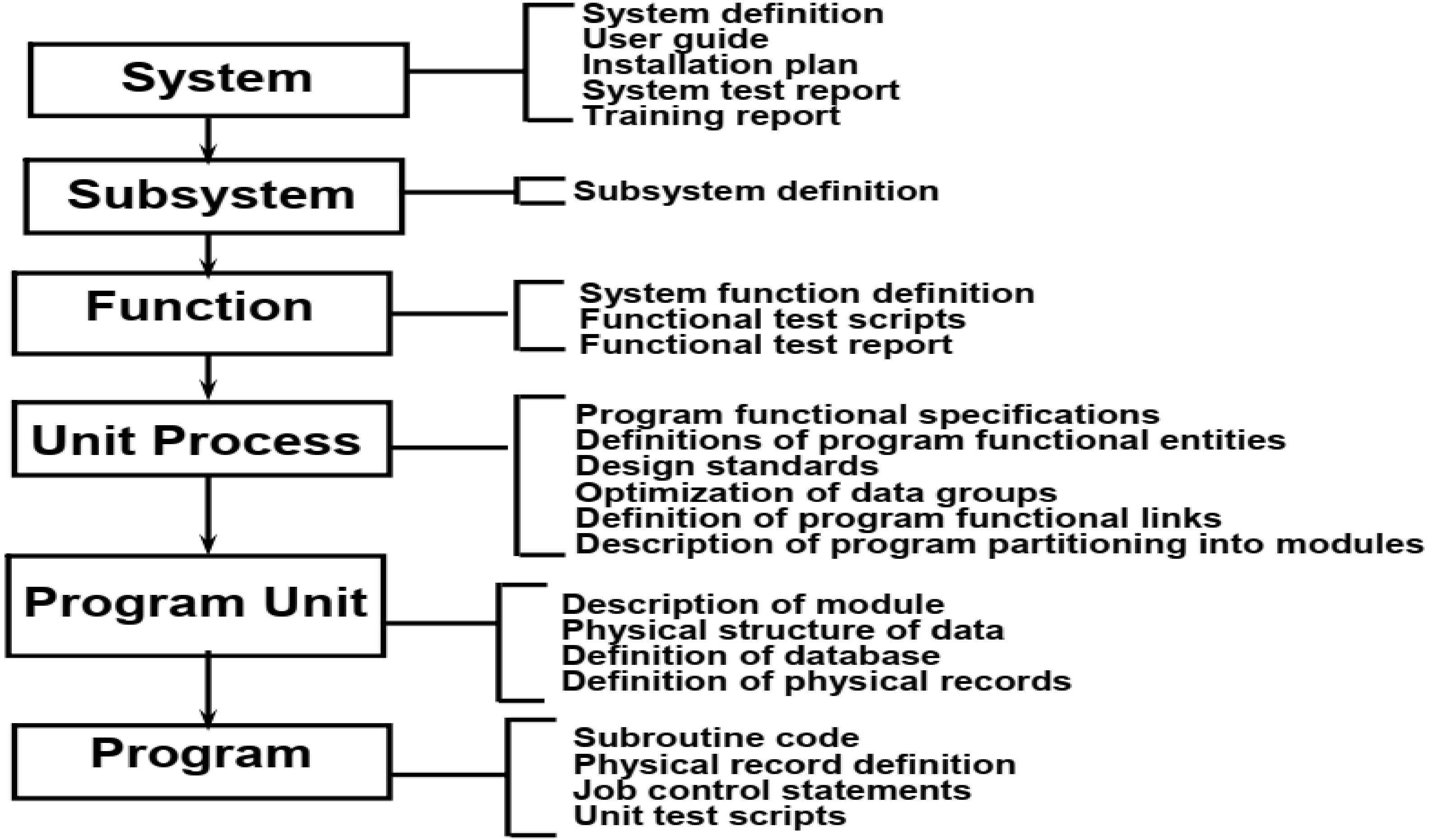
Plan and Establish Baseline

How To Select Configuration Items

- Configuration Items are selected by the SCM personnel assigned to projects and project teams during the design process by applying a set of selection criteria to the elements of the decomposed system.

- Configuration Item selection criteria may include a number of factors, some of which may be unique to the project.
  - Grouping together similar functional requirements or processes.
  - Grouping processes that can be developed and tested by a single group, or as a unit.
  - Separating requirements or processes that must be completed at different times.
  - Separating requirements or processes that must run on different hardware platforms.

Components of a Configuration Item



System
- System definition
- User guide
- Installation plan
- System test report
- Training report

Subsystem
- Subsystem definition

Function
- System function definition
- Functional test scripts
- Functional test report

Unit Process
- Program functional specifications
- Definitions of program functional entities
- Design standards
- Optimization of data groups
- Definition of program functional links
- Description of program partitioning into modules

Program Unit
- Description of module
- Physical structure of data
- Definition of database
- Definition of physical records

Program
- Subroutine code
- Physical record definition
- Job control statements
- Unit test scripts

Baseline Management

- Baseline control

  SCM must ensure baseline control while providing service to software developers. The baseline must be protected against unauthorized change, at the same time developers should be able to work on their code (i.e., modify, tests).

- Baseline management provides:
  - Project management/customer visibility of the project
  - Complete development history
  - Formal review and audit points
  - Improves system control

Plan & Establish Baseline

- □ Functional Baseline – Requirements that describe the system's function agreed to by customer and project managers.

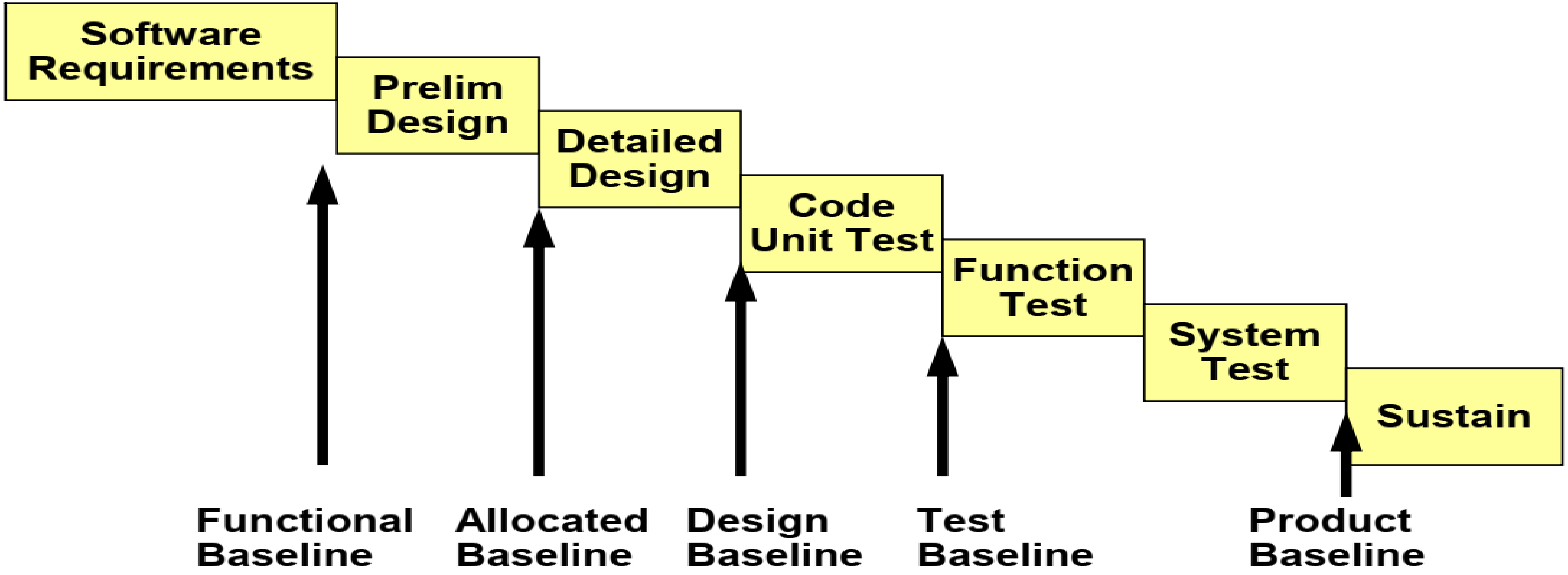
- □ Allocated Baseline – Customer approved documents assigning (allocating) functional requirements to specific CIs.

- □ *Design Baseline* – Point at which detailed system design is complete.

- □ *Test Baseline* – point at which tests are run to verify requirements are met and ready for release.

- □ Product Baseline – the release-ready collection of CIs.

Establishing Baselines



Configuration Management Library

- □ **SCM Library Definition**:

  A controlled collection of software and related documentation designed to aid in software development, use, or maintenance.

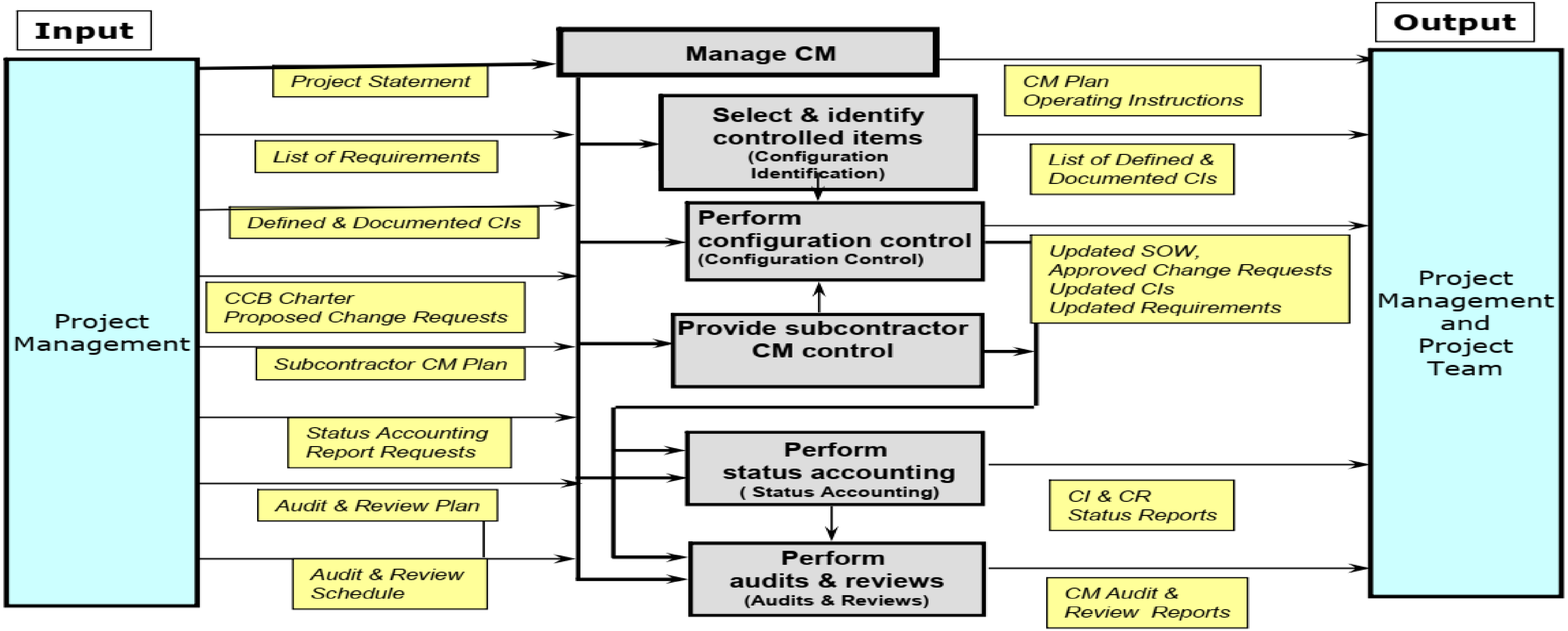
- □ **Purpose of the SCM Library:**

  A controlled repository of CIs and CI components to provide the staff with definition of the hardware, or software tools and procedures that will enable the re-creation of any version of any representation of software products.

Configuration Management Libraries

- □ Working or Development Library
  - ■ Programmer's assigned work area for code generation and testing (developer's play pen).
- □ Software Configuration Management Library
  - ■ Software Configuration Management controlled, for example:
    - □ Partitioned into development, test, freeze environments
- □ Archive Library, Reuse Library
  - ■ An area which provides for the storage of software and documentation for reuse or storage.

Software Configuration Management Process



Input

Project Management

Project Statement

List of Requirements

Defined & Documented CIs

CCB Charter
Proposed Change Requests

Subcontractor CM Plan

Status Accounting
Report Requests

Audit & Review Plan

Audit & Review
Schedule

Manage CM

Select & identify controlled items
(Configuration Identification)

Perform configuration control
(Configuration Control)

Provide subcontractor CM control

Perform status accounting
(Status Accounting)

Perform audits & reviews
(Audits & Reviews)

CM Plan
Operating Instructions

List of Defined &
Documented CIs

Updated SOW,
Approved Change Requests
Updated CIs
Updated Requirements

CI & CR
Status Reports

CM Audit &
Review Reports

Output

Project Management and Project Team

Summary

- Change management is one of the fundamental activities of software engineering.

- Configuration management helps control changes and coordinates the work products of many project team members.

- The baseline is the official source for code and the repository for all completed work, it is the official standard which subsequent work is based and which authorized changes are made.

- The baseline contains the project code and is stored in the SCM library.