CHANGE MANAGEMENT

- Introduction
- SCM repository
- The SCM process

INTRODUCTION

What is Change Management

- Also called software configuration management (SCM)

- It is an **umbrella activity** that is applied throughout the software process

- It's goal is to maximize productivity by minimizing mistakes caused by confusion when coordinating software development

- SCM identifies, organizes, and controls modifications to the software being built by a software development team

- SCM activities are formulated to identify change, control change, ensure that change is being properly implemented, and report changes to others who may have an interest

What is Change Management (continued)

- SCM is initiated when the project begins and terminates when the software is taken out of operation

- View of SCM from various roles
  - *Project manager -> an auditing mechanism*

  - *SCM manager -> a controlling, tracking, and policy making mechanism*

  - *Software engineer -> a changing, building, and access control mechanism*

  - *Customer -> a quality assurance and product identification mechanism*

Software Configuration

■ The Output from the software process makes up the software configuration

- *Computer programs (both source code files and executable files)*
- *Work products that describe the computer programs (documents targeted at both technical practitioners and users)*
- *Data (contained within the programs themselves or in external files)*

■ The major danger to a software configuration is **change**

- *First Law of System Engineering: "No matter where you are in the system life cycle, the system will change, and the desire to change, it will persist throughout the life cycle"*

Origins of Software Change

- Errors detected in the software need to be corrected

- New business or market conditions dictate changes in product requirements or business rules

- New customer needs demand modifications of data produced by information systems, functionality delivered by products, or services delivered by a computer-based system

- Reorganization or business growth/downsizing causes changes in project priorities or software engineering team structure

- Budgetary or scheduling constraints cause a redefinition of the system or product

Elements of a Configuration Management System

- Configuration elements
  - *A set of tools coupled with a file management (e.g., database) system that enables access to and management of each software configuration item*

- Process elements
  - *A collection of procedures and tasks that define an effective approach to change management for all participants*
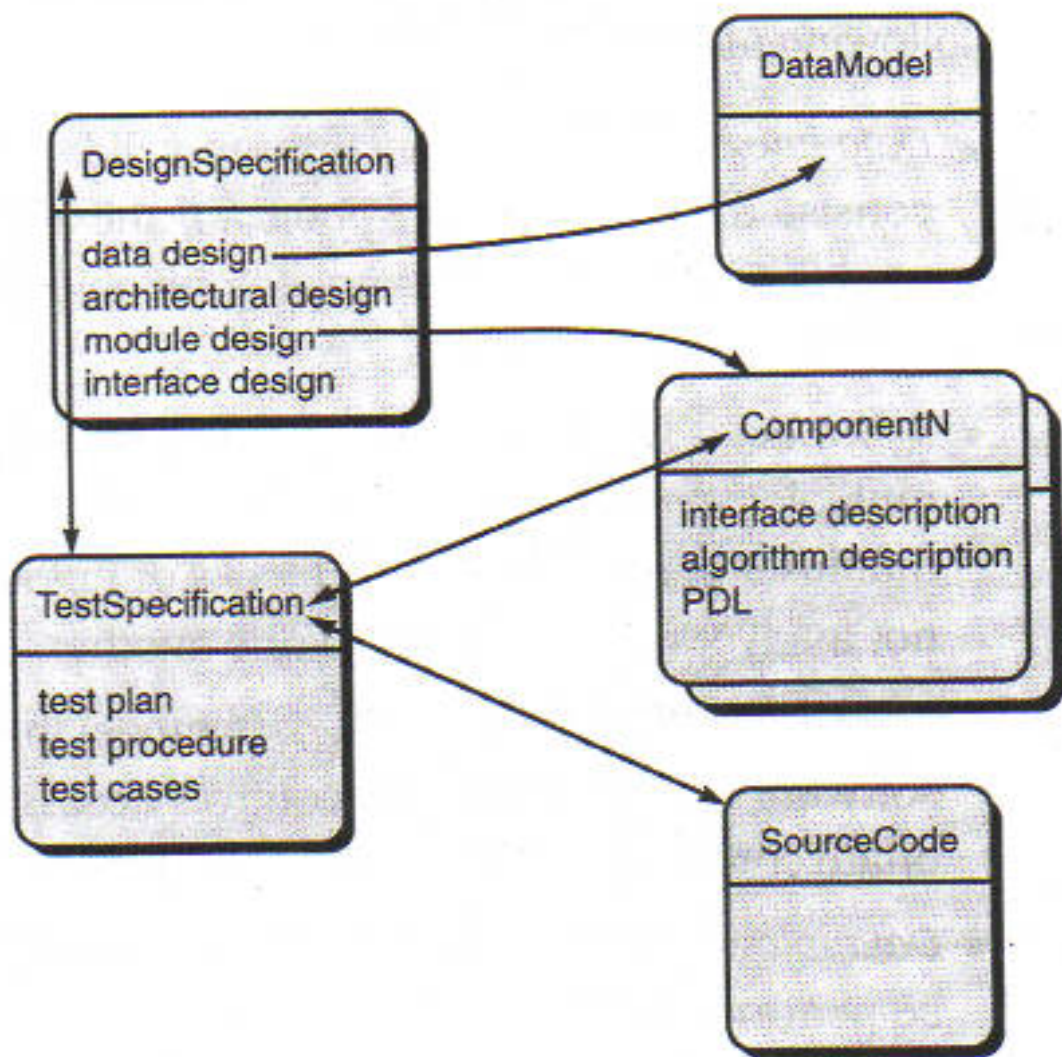
- Construction elements
  - *A set of tools that automate the construction of software by ensuring that the proper set of valid components (i.e., the correct version) is assembled*
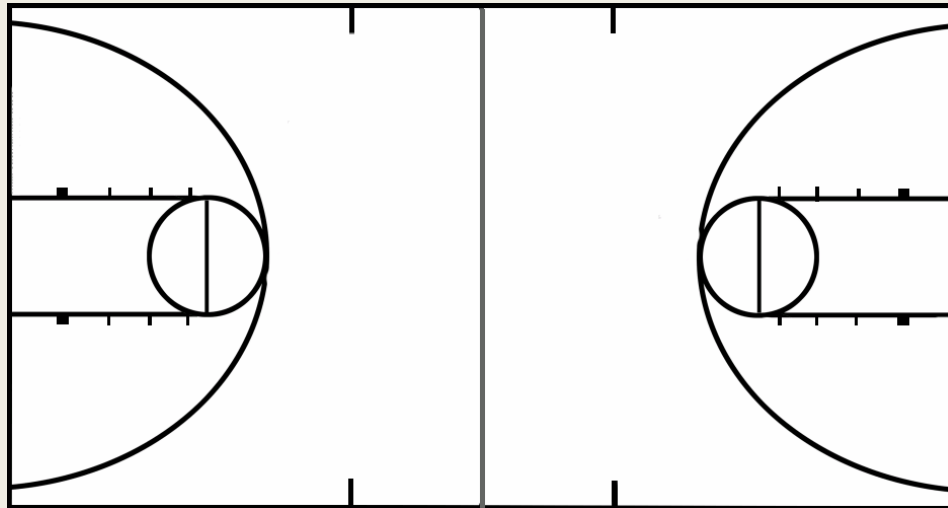
- Human elements
  - *A set of tools and process features used by a software team to implement effective SCM*

Software Configuration Items (SCIs)

- Element of information that can be as small as single UML diagram or as large as the complete design document.

- Can be considered as a single section of a large specification or one test case in a large suit of test cases.

- SCIs are stored in the database as a Configuration Object with name, attributes and is connected to other objects by relationships.

DataModel

DesignSpecification

data design
architectural design
module design
interface design

ComponentN

interface description
algorithm description
PDL

TestSpecification

test plan
test procedure
test cases

SourceCode

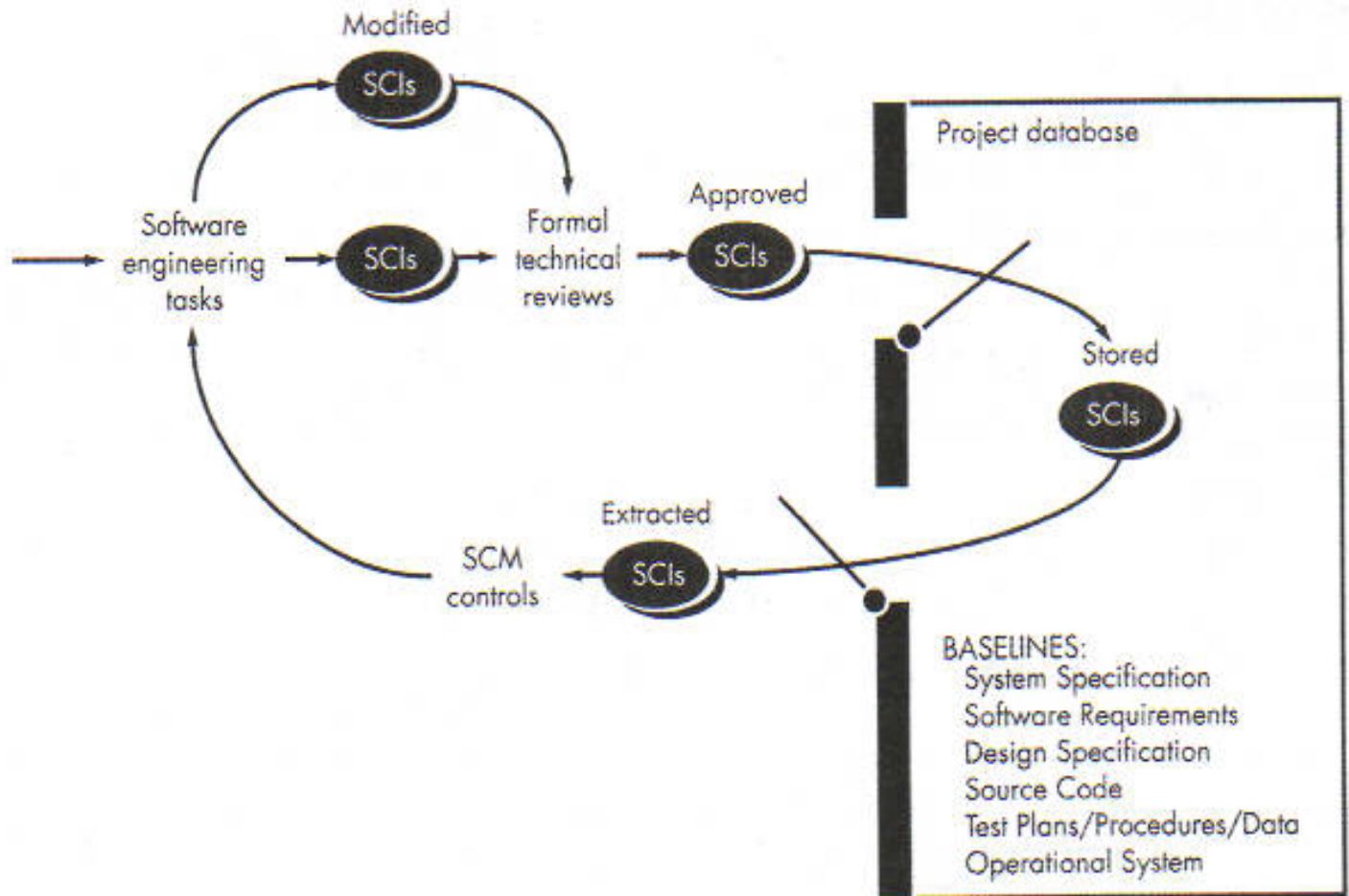# Have you established a baseline yet?

Baseline

- An SCM concept that helps practitioners to control change without seriously impeding justifiable change

- IEEE Definition: **A specification or product that has been formally reviewed and agreed upon, and that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures**

- It is a milestone in the development of software and is marked by the delivery of one or more computer software configuration items (SCIs) that have been approved as a consequence of a formal technical review

- A SCI may be such work products as a document (as listed in MIL-STD-498), a test suite, or a software component

Baselining Process

1) A series of software engineering tasks produces a SCI

2) The SCI is reviewed and possibly approved

3) The approved SCI is given a new version number and placed in a project database (i.e., software repository)

4) A copy of the SCI is taken from the project database and examined/modified by a software engineer

5) The baselining of the modified SCI goes back to Step #2

Modified

SCIs

Software engineering tasks

SCIs

Formal technical reviews

Approved

SCIs

Project database

Stored

SCIs

Extracted

SCIs

SCM controls

SCIs

BASELINES:
System Specification
Software Requirements
Design Specification
Source Code
Test Plans/Procedures/Data
Operational System

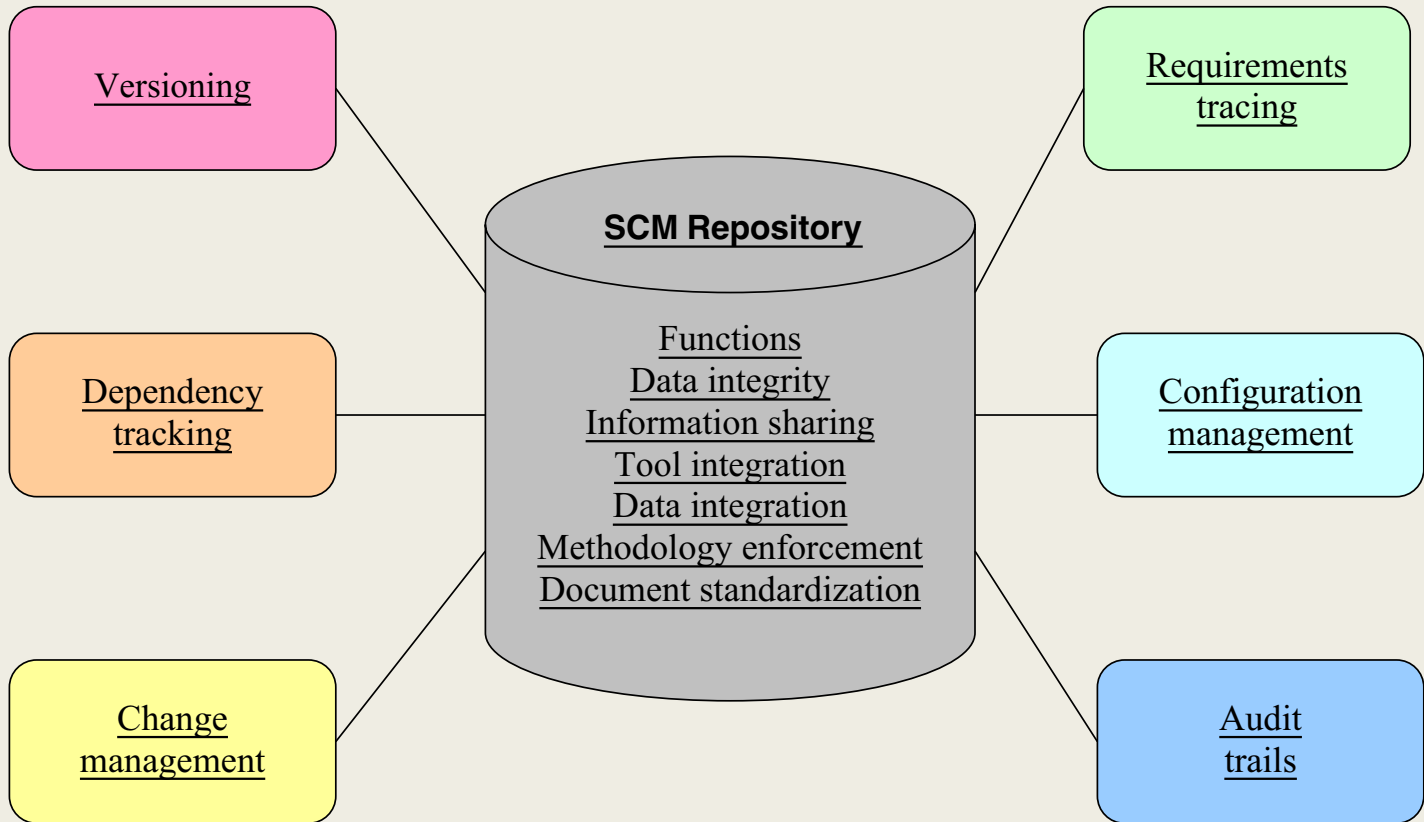# THE SCM REPOSITORY

Paper-based vs. Automated Repositories

Problems with paper-based repositories (i.e., file cabinet containing folders)

- *Finding a configuration item when it was needed was often difficult*
- *Determining which items were changed, when and by whom was often challenging*
- *Constructing a new version of an existing program was time consuming and error prone*
- *Describing detailed or complex relationships between configuration items was virtually impossible*

Today's automated SCM repository

- *It is a set of mechanisms and data structures that allow a software team to manage change in an effective manner*
- *It acts as the center for both accumulation and storage of software engineering information*
- *Software engineers use tools integrated with the repository to interact with it*

Automated SCM Repository (Functions and Tools)

Versioning

Requirements tracing

Dependency tracking

SCM Repository

Functions
Data integrity
Information sharing
Tool integration
Data integration
Methodology enforcement
Document standardization

Configuration management

Change management

Audit trails

Functions of an SCM Repository

- Data integrity
    - *Validates entries, ensures consistency, cascades modifications*

- Information sharing
    - *Shares information among developers and tools, manages and controls multi-user access*

- Tool integration
    - *Establishes a data model that can be accessed by many software engineering tools, controls access to the data*

- Data integration
    - *Allows various SCM tasks to be performed on one or more CSCIs*

- Methodology enforcement
    - *Defines an entity-relationship model for the repository that implies a specific process model for software engineering*

- Document standardization
    - *Defines objects in the repository to guarantee a standard approach for creation of software engineering documents*

Toolset Used on a Repository

- Versioning
    - *Save and retrieve all repository objects based on version number*

- Dependency tracking and change management
    - *Track and respond to the changes in the state and relationship of all objects in the repository*

- Requirements tracing
    - *(Forward tracing) Track the design and construction components and deliverables that result from a specific requirements specification*
    - *(Backward tracing) Identify which requirement generated any given work product*

- Configuration management
    - *Track a series of configurations representing specific project milestones or production releases*

- Audit trails
    - *Establish information about when, why, and by whom changes are made in the repository*

THE SCM PROCESS

Primary Objectives of the SCM Process

- Identify all items that collectively define the software configuration

- Manage changes to one or more of these items

- Facilitate construction of different versions of an application

- Ensure the software quality is maintained as the configuration evolves over time

- Provide information on changes that have occurred

SCM Questions

How does a software team identify the discrete elements of a software configuration?

How does an organization manage the many existing versions of a program (and its documentation) in a manner that will enable change to be accommodated efficiently?

How does an organization control changes before and after software is released to a customer?

Who has responsibility for approving and ranking changes?

How can we ensure that changes have been made properly?

What mechanism is used to appraise others of changes that are made?

SCM Tasks

Status reporting

Configuration auditing

Version control

Change control

Identification

SCI   SCI
SCI   SCI
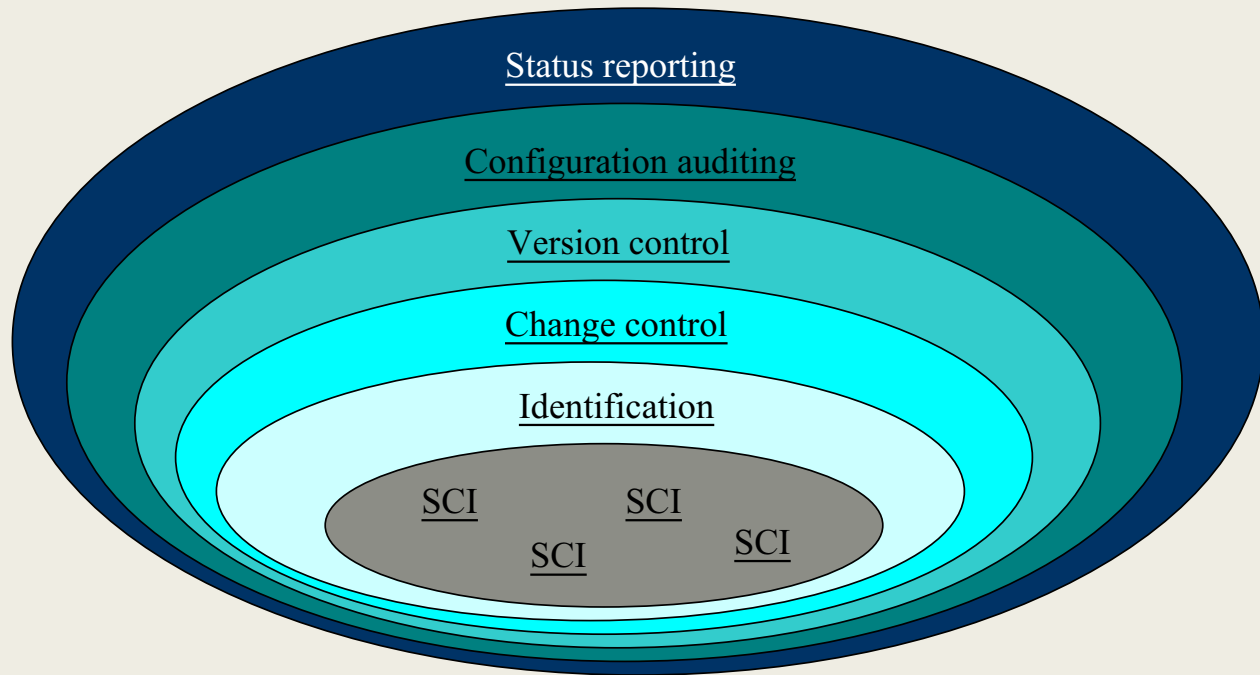
SCM Tasks (continued)

- Concentric layers (from inner to outer)
  - *Identification*
  - *Change control*
  - *Version control*
  - *Configuration auditing*
  - *Status reporting*

- SCIs flow outward through these layers during their life cycle

- SCIs ultimately become part of the configuration of one or more versions of a software application or system

Identification Task

■ Identification separately names each SCI and then organizes it in the SCM repository using an object-oriented approach

■ Objects start out as basic objects and are then grouped into aggregate objects

■ Each object has a set of distinct features that identify it

   – *A name that is unambiguous to all other objects*

   – *A description that contains the SCI type, a project identifier, and change and/or version information*

   – *List of resources needed by the object*

   – *The object realization (i.e., the document, the file, the model, etc.)*

Change Control Task

- Change control is a procedural activity that ensures quality and consistency as changes are made to a configuration object

- A change request is submitted to a configuration control authority, which is usually a change control board (CCB)
    - *The request is evaluated for technical merit, potential side effects, overall impact on other configuration objects and system functions, and projected cost in terms of money, time, and resources*

- An engineering change order (ECO) is issued for each approved change request
    - *Describes the change to be made, the constraints to follow, and the criteria for review and audit*

- The baselined SCI is obtained from the SCM repository
    - *Access control governs which software engineers have the authority to access and modify a particular configuration object*
    - *Synchronization control helps to ensure that parallel changes performed by two different people don't overwrite one another*

Version Control Task

- Version control is a set of procedures and tools for managing the creation and use of multiple occurrences of objects in the SCM repository

- Required version control capabilities
  - *An SCM repository that stores all relevant configuration objects*
  - *A version management capability that stores all versions of a configuration object (or enables any version to be constructed using differences from past versions)*
  - *A make facility that enables the software engineer to collect all relevant configuration objects and construct a specific version of the software*
  - *Issues tracking (bug tracking) capability that enables the team to record and track the status of all outstanding issues associated with each configuration object*

- The SCM repository maintains a change set
  - *Serves as a collection of all changes made to a baseline configuration*
  - *Used to create a specific version of the software*
  - *Captures all changes to all files in the configuration along with the reason for changes and details of who made the changes*

Configuration Auditing Task

■   Configuration auditing is an SQA activity that helps to ensure that quality is maintained as changes are made

■   It complements the formal technical review and is conducted by the SQA group

■   It addresses the following questions
   –   *Has the change specified in the ECO been made? Have any additional modifications been incorporated?*
   –   *Has a formal technical review been conducted to assess technical correctness?*
   –   *Has the software process been followed, and have software engineering standards been properly applied?*
   –   *Has the change been "highlighted" and "documented" in the SCI? Have the change data and change author been specified? Do the attributes of the configuration object reflect the change?*
   –   *Have SCM procedures for noting the change, recording it, and reporting it been followed?*
   –   *Have all related SCIs been properly updated?*

■   A configuration audit ensures that
   –   *The correct SCIs (by version) have been incorporated into a specific build*
   –   *That all documentation is up-to-date and consistent with the version that has been built*

Status Reporting Task

- Configuration status reporting (CSR) is also called status accounting

- Provides information about each change to those personnel in an organization with a need to know

- Answers what happened, who did it, when did it happen, and what else will be affected?

- Sources of entries for configuration status reporting
  - *Each time a SCI is assigned new or updated information*
  - *Each time a change is approved by the CCB and an ECO is issued*
  - *Each time a configuration audit is conducted*

- The configuration status report
  - *Placed in an on-line database or on a website for software developers and maintainers to read*
  - *Given to management and practitioners to keep them appraised of important changes to the project SCIs*

Summary

- Introduction
- SCM Repository
- SCM Process
  - *Identification*
  - *Change control*
  - *Version control*
  - *Configuration auditing*
  - *Status reporting*