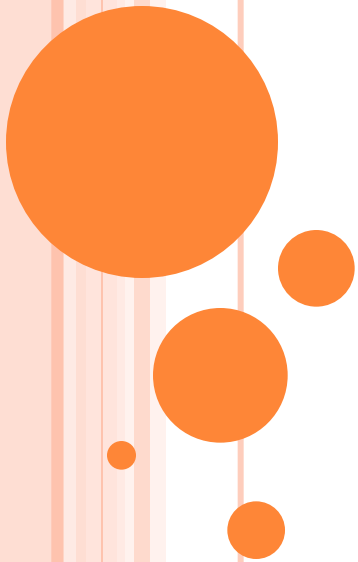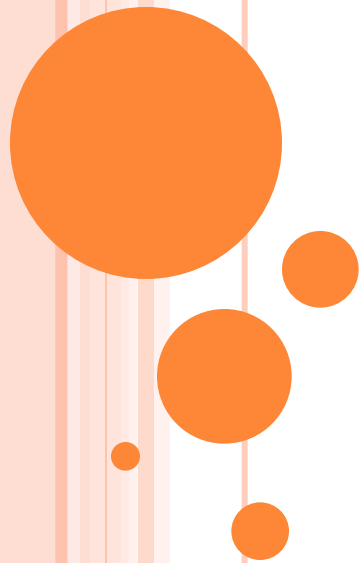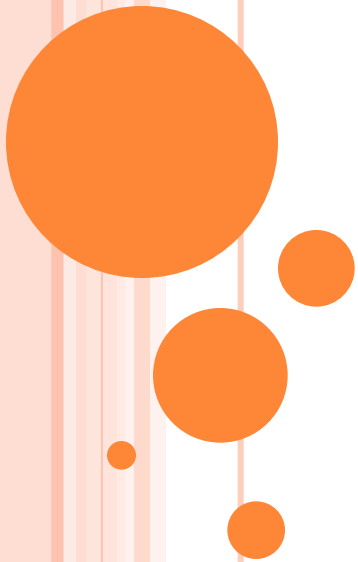# SOFTWARE ENGINEERING

LECTURE-15
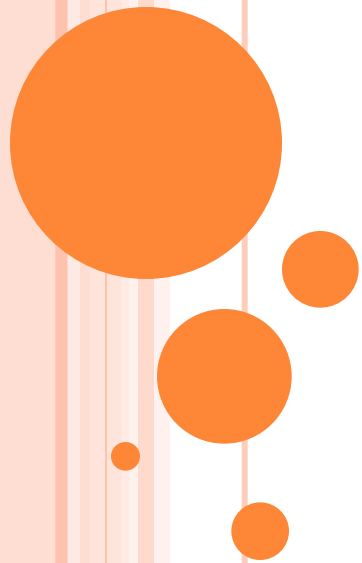
Change Management

TOPICS COVERED

- **Introduction**
- **SCM repository**
- **The SCM process**

INTRODUCTION

WHAT IS CHANGE MANAGEMENT

- Also called software configuration management (SCM)
- It is an umbrella activity that is <u>applied throughout</u> the software process
- It's goal is to <u>maximize</u> productivity by <u>minimizing</u> mistakes caused by confusion when <u>coordinating</u> software development
- SCM identifies, organizes, and controls <u>modifications</u> to the software being built by a software development team
- SCM activities are formulated to <u>identify</u> change, <u>control</u> change, <u>ensure</u> that change is being properly implemented, and <u>report</u> changes to others who may have an interest

5

(More on next slide)

WHAT IS CHANGE MANAGEMENT (CONTINUED)

- SCM is <u>initiated</u> when the project <u>begins</u> and <u>terminates</u> when the software is taken <u>out of operation</u>
- <u>View</u> 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 <u>Output</u> from the software process makes up the software configuration
  - <u>Computer programs</u> (both source code files and executable files)
  - <u>Work products</u> that describe the computer programs (documents targeted at both technical practitioners and users)
  - <u>Data</u> (contained within the programs themselves or in external files)

- The major <u>danger</u> to a software configuration is <u>change</u>
  - First Law of System Engineering: "No matter where you are in the system life cycle, <u>the system will change</u>, and the desire to change it will persist throughout the life cycle"

ORIGINS OF SOFTWARE CHANGE

- <u>Errors</u> detected in the software need to be corrected
- <u>New business or market conditions</u> dictate changes in product requirements or business rules
- <u>New customer needs</u> demand modifications of data produced by information systems, functionality delivered by products, or services delivered by a computer-based system
- <u>Reorganization or business growth/downsizing</u> causes changes in project priorities or software engineering team structure
- <u>Budgetary or scheduling constraints</u> cause a redefinition of the system or product

ELEMENTS OF A CONFIGURATION MANAGEMENT SYSTEM

- Configuration elements
  - A set of <u>tools</u> coupled with a <u>file management</u> (e.g., database) system that enables <u>access</u> to and <u>management</u> of each software configuration item
- Process elements
  - A collection of <u>procedures</u> and <u>tasks</u> that define an effective <u>approach</u> to change management for all participants
- Construction elements
  - A set of <u>tools</u> that automate the <u>construction</u> of software by ensuring that the proper set of valid components (i.e., the correct version) is <u>assembled</u>
- Human elements
  - A set of <u>tools</u> and <u>process features</u> used by a software team to <u>implement</u> effective SCM

# Have you established a <u>baseline</u> yet?
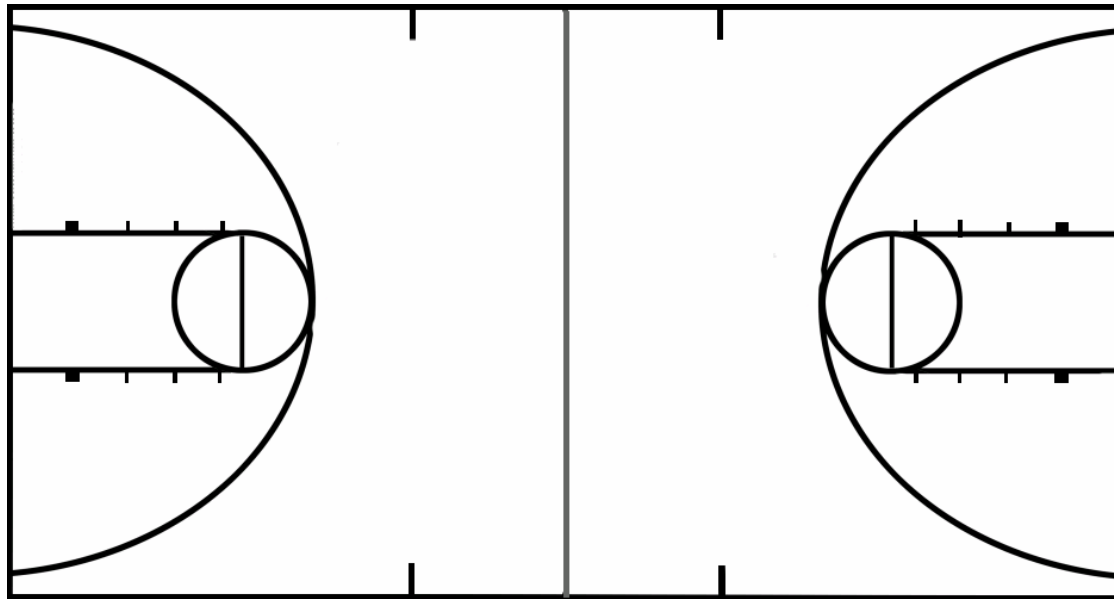
BASELINE

- An SCM concept that helps <u>practitioners</u> to control change without seriously <u>impeding</u> justifiable change

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

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

- A CSCI may be such work products as a <u>document</u> (as listed in MIL-STD-498), a <u>test suite</u>, or a <u>software component</u>

Baselining Process

1) A series of software engineering tasks <u>produces</u> a CSCI

2) The CSCI is <u>reviewed</u> and possibly <u>approved</u>

3) The approved CSCI is given a new <u>version number</u> and placed in a <u>project database</u> (i.e., software repository)

4) A <u>copy</u> of the CSCI is <u>taken</u> from the project database and <u>examined/modified</u> by a software engineer

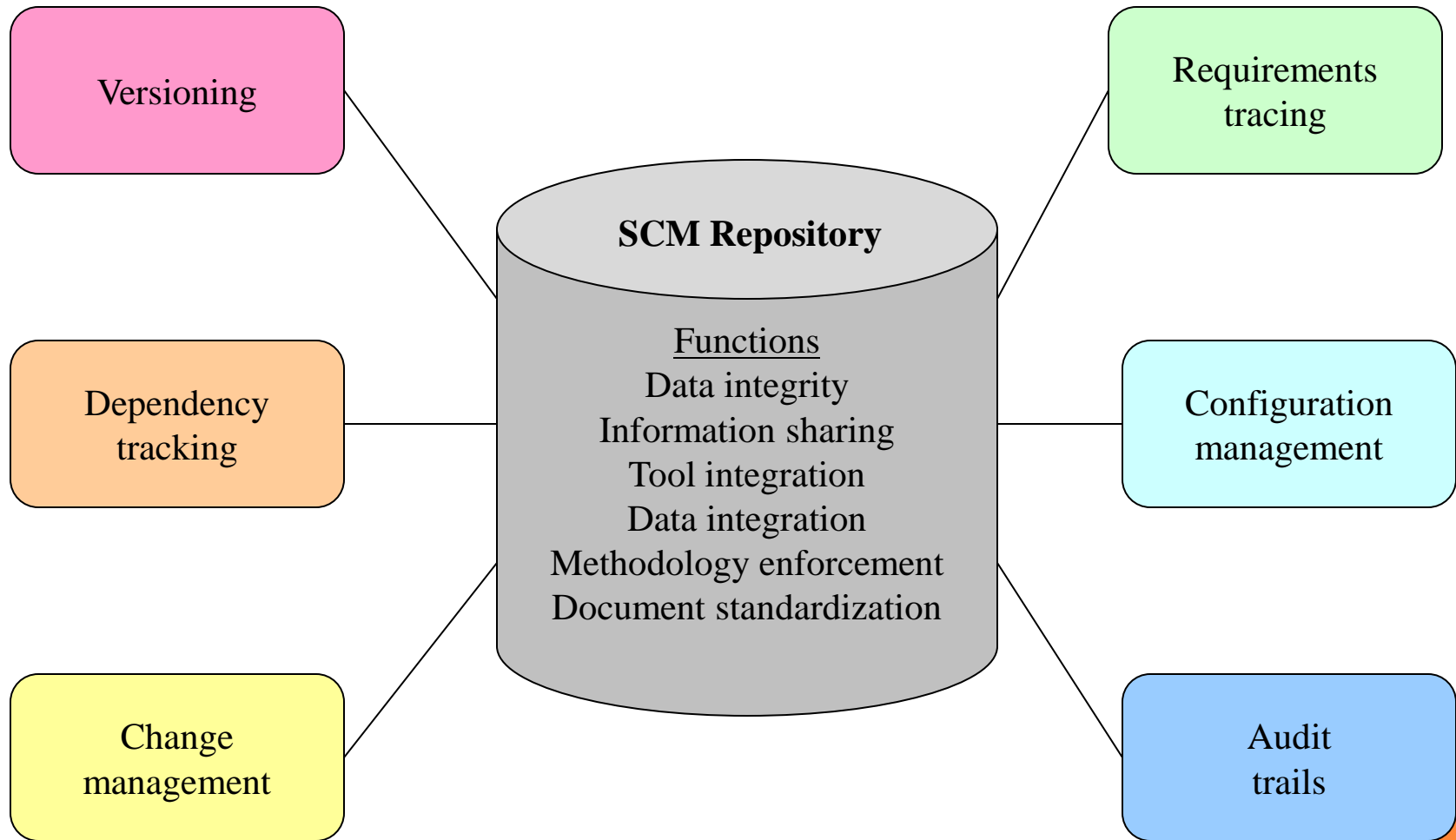5) The baselining of the <u>modified</u> CSCI goes back to Step #2

THE SCM REPOSITORY
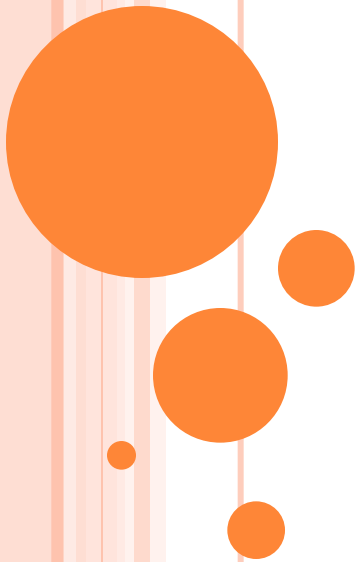
Paper-based vs. Automated Repositories

- Problems with <u>paper-based</u> repositories (i.e., file cabinet containing folders)
  - Finding a configuration item when it was needed was often <u>difficult</u>
  - Determining which items were changed, when and by whom was often challenging
  - Constructing a new version of an existing program was <u>time consuming</u> and <u>error prone</u>
  - Describing detailed or complex relationships between configuration items was <u>virtually impossible</u>
- Today's <u>automated</u> SCM repository
  - It is a set of mechanisms and data structures that allow a software team to <u>manage change</u> in an <u>effective</u> manner
  - It acts as the <u>center</u> for both <u>accumulation</u> and <u>storage</u> of software engineering information
  - Software engineers use tools <u>integrated</u> with the repository to interact with it

14

AUTOMATED SCM REPOSITORY (FUNCTIONS AND TOOLS)

Versioning

Requirements tracing

SCM Repository

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

Dependency tracking

Configuration management

Change management

Audit trails

(Explained on next two slides)

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
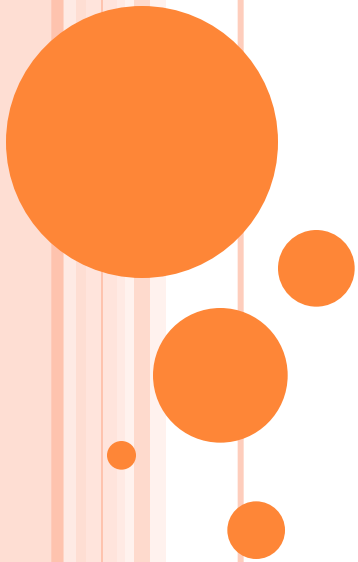
Toolset Used on a Repository

- 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

SUMMARY OF CM TOOLS

- http://www.daveeaton.com/scm/CMTools.html

- http://www.laatuk.com/tools/SCM_tools.html

- http://www.snuffybear.com/ucmcentral_new_vendorlinks.htm

- http://www.google.com/Top/Computers/Software/Configuration_Management/Tools/

- http://stason.org/TULARC/business/config-version-management-tools/69-CM-Tools-With-World-Wide-Web-Sites.html

- http://www.cmcrossroads.com/cm-resources/tools/commercial-cm-tools

The SCM Process

PRIMARY OBJECTIVES OF THE SCM PROCESS

- <u>Identify all items</u> that collectively define the software configuration

- <u>Manage changes</u> to one or more of these items

- <u>Facilitate</u> construction of different <u>versions</u> of an application

- <u>Ensure</u> the software <u>quality</u> is <u>maintained</u> as the configuration evolves over time

- <u>Provide</u> information on changes that have occurred

(Compare this process to the five SCM tasks)

SCM QUESTIONS

- How does a software team <u>identify</u> the discrete elements of a software configuration?

- How does an organization <u>manage</u> 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 <u>control</u> changes before and after software is released to a customer?

- Who has responsibility for <u>approving</u> and ranking <u>changes</u>?

- How can we <u>ensure</u> that changes have been made properly?

- What mechanism is used to <u>appraise</u> others of changes that are made?

SCM Tasks

Status reporting

Configuration auditing

Version control

Change control

Identification

CSCI   CSCI

CSCI   CSCI
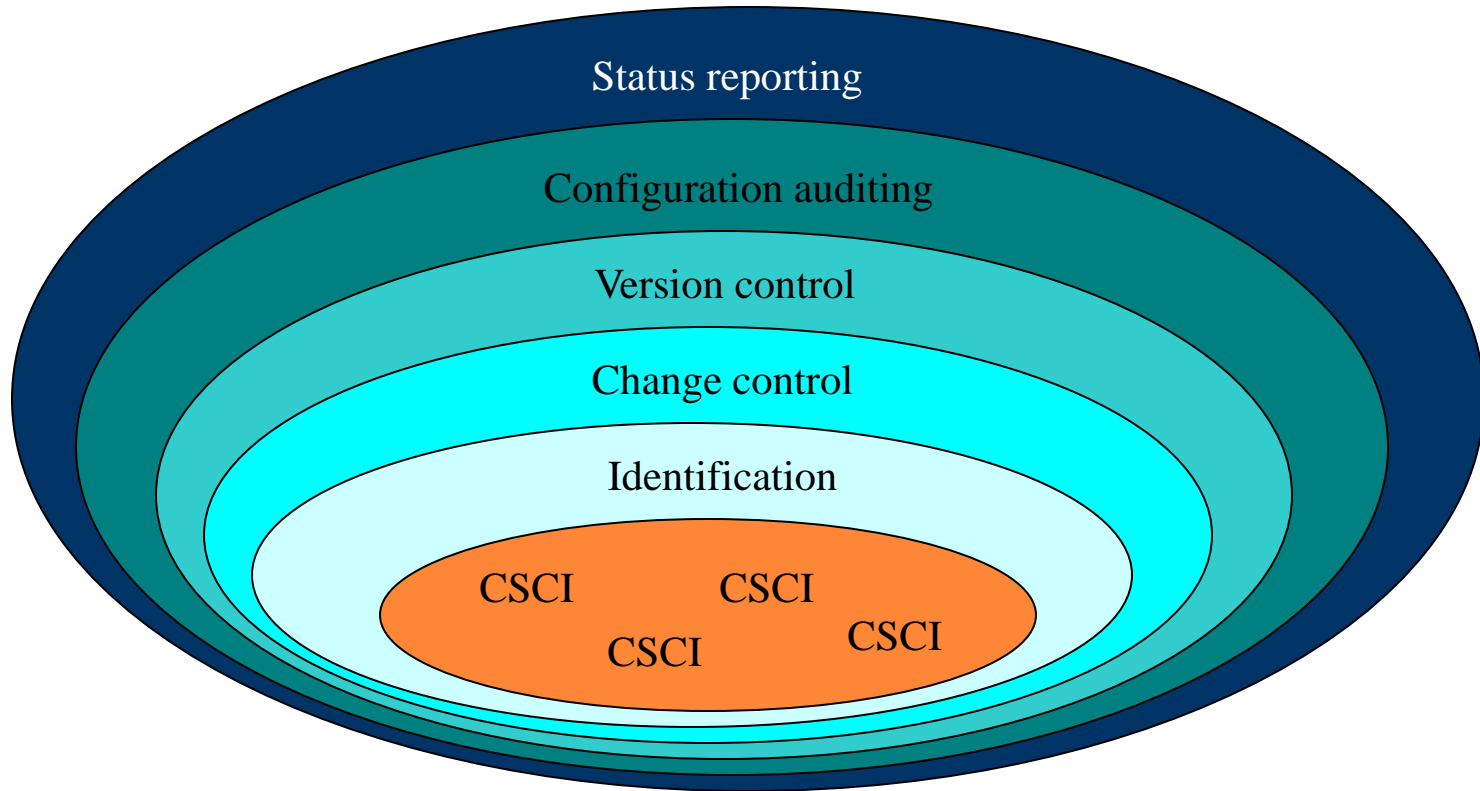
(More on next slide)

SCM Tasks (continued)

- Concentric layers (from inner to outer)
  - Identification
  - Change control
  - Version control
  - Configuration auditing
  - Status reporting
- CSCIs flow <u>outward</u> through these layers during their life cycle
- CSCIs ultimately become part of the configuration of one or more <u>versions</u> of a software application or system

23

IDENTIFICATION TASK

- Identification <u>separately names</u> each CSCI and then <u>organizes</u> it in the SCM repository using an object-oriented approach
- Objects start out as basic objects and are then <u>grouped</u> into aggregate objects
- Each object has a set of <u>distinct features</u> that identify it
  - A <u>name</u> that is unambiguous to all other objects
  - A <u>description</u> that contains the CSCI type, a project identifier, and change and/or version information
  - List of <u>resources</u> needed by the object
  - The <u>object realization</u> (i.e., the document, the file, the model, etc.)

24

CHANGE CONTROL TASK

- Change control is a <u>procedural</u> activity that ensures quality and consistency as changes are made to a configuration object
- A change request is <u>submitted</u> to a configuration control authority, which is usually a change control board (CCB)
  - The request is <u>evaluated</u> 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 <u>issued</u> for each <u>approved</u> change request
  - Describes the <u>change</u> to be made, the constraints to follow, and the <u>criteria</u> for review and audit
- The baselined CSCI is <u>obtained</u> from the SCM repository
  - <u>Access control</u> governs <u>which</u> software engineers have the authority to access and modify a particular configuration object
  - <u>Synchronization control</u> helps to ensure that <u>parallel</u> 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 <u>multiple occurrences</u> of objects in the SCM repository

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

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

26

CONFIGURATION AUDITING TASK

- Configuration auditing is an SQA activity that helps to ensure that <u>quality is maintained</u> as changes are made
- It complements the <u>formal technical review</u> and is conducted by the SQA group
- It addresses the following questions
  - Has the change specified in the ECO been <u>made</u>? Have any additional modifications been incorporated?
  - Has a formal technical review been <u>conducted</u> to assess technical correctness?
  - Has the software process been <u>followed</u>, and have software engineering standards been properly <u>applied</u>?
  - Has the change been "<u>highlighted</u>" and "<u>documented</u>" in the CSCI? Have the change data and change author been <u>specified</u>? Do the attributes of the configuration object <u>reflect</u> the change?
  - Have SCM procedures for noting the change, recording it, and reporting it been <u>followed</u>?
  - Have all related CSCIs been properly <u>updated</u>?
- A configuration audit ensures that
  - The correct CSCIs (by version) have been <u>incorporated</u> into a specific build
  - That all documentation is <u>up-to-date</u> and <u>consistent</u> with the version that has been built

27

Status Reporting Task

- Configuration status reporting (CSR) is also called <u>status accounting</u>
- <u>Provides information</u> about each change to those personnel in an organization with a need to know
- Answers <u>what</u> happened, <u>who</u> did it, <u>when</u> did it happen, and <u>what else</u> will be affected?
- Sources of <u>entries</u> for configuration status reporting
  - Each time a CSCI is <u>assigned</u> new or updated information
  - Each time a change is <u>approved</u> by the CCB and an ECO is issued
  - Each time a configuration audit is <u>conducted</u>
- The configuration status report
  - Placed in an <u>on-line database</u> or on a website for software developers and maintainers to read
  - Given to <u>management and practitioners</u> to keep them appraised of important changes to the project CSCIs

SUMMARY

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

☺