Software Configuration Management

- **Software Configuration**: All items that constitute the software while under the development (e.g., programs, data, documents such as the software requirements specification, test cases, etc.); they are referred to as Software Configuration Items or SCIs

- Most (or all) SCIs change during the development; such changes must be effectively controlled

For exclusive use of Professor Saiedian's students
Software Configuration Management (continued)

Software Configuration Management (SCM) is the process of identifying and defining the SCIs in the system and coordinating the changes made to these items. A formal definition:

“SCM is the process of identifying and defining the items in the system, controlling the change of these items throughout their life cycle, recording and reporting the status of items and change requests, and verifying the completeness and correctness of items”

IEEE 1987
Software Configuration Management
(continued)

SCM provides for systematic evolution of a software under development and provides for:

- visibility
- controlled change
- traceability
- monitoring
CMM on Software Configuration Management

SCM and the CMM. Goals of Level 2 SCM KPA:

- Goal 1: Software configuration management activities are planned
- Goal 2: Selected software work products are identified, controlled, and available
- Goal 3: Changes to identified software work products are controlled
- Goal 4: Affected groups and individuals are informed of the status and content of software baselines

For exclusive use of Professor Saiedian’s students
Configuration Management

Primary Task: To manage change

Baseline. “A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedure.” IEEE 1990

For exclusive use of Professor Saiedian’s students
Components of the SCM

Major components of the SCM:

- Software Configuration identification
- Change Control
- Configuration Status Accounting
- Configuration Audit
Software Configuration

Identification

First Step in SCM: Identification of the SCIs
No fast rules: once a SCI has been identified, it should be given an identification name

Examples of the SCIs:

- The entire product of a software development phase (e.g., a requirements document)
- Chapters or sections of a document
- A separately compilable module
- A file consisting of a number of modules
- A file consisting of module definitions
Baselines of a Typical Development Life Cycle

- Phase/Discipline: Requirements Analysis
  Baseline: Software Requirements Specification

- Phase/Discipline: Software Design
  Baseline: Design Specification

- Phase/Discipline: Coding and Implementation
  Baseline: Source Code

- Phase/Discipline: Testing and Integration
  Baseline: Test Plans and Data

- Phase/Discipline: Acceptance Testing/Release
  Baseline: Operational Software

For exclusive use of Professor Saiedian’s students
Change Control

- Change control is the process of controlling the changes to the SCIs. A formal definition:

  "An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification."

  IEEE 1990
Change Control Process

Decision making process is carried out by the Configuration Control Board (CCB) which consists of one or more individuals led by the Configuration Manager (CM), responsible for configuration management.

The process is as follows:

1. While a SCI is under development (working state), it can changed freely since it is not under SCM yet.
2. When a SCI is in a stable condition, it is submitted to the CM for review (under review state).
(3) While under review, the SCI is considered “frozen”

(4) Once the SCI is reviewed (for satisfactory quality) and approved, it is entered into a library - the SCI is now formally under SCM:

(5) Once in the library (i.e., under the SCM), the SCI cannot be changed unless the change is approved by the CCB
(6) Request for change must be formally initiated via a Change Request (CR) form. A CR form has three components:

6.1 Change Request Proposal

- Proposer information
- Description of change
- Reason for the change
- SCIs affected by the change
- Priority of the change (fault error?)
Change Control Process (continued)

6.2 CCB’s Decision (completed by CM)

- CCB’s approval/disapproval decision (based on cost/benefit analysis, quality issues, scheduling)
- CCB assigned unique number to the proposed change for future references
- Actions recommended by the CM (e.g., no action, change the SCI, changes to programs, changes to the documents)

6.3 Implementation of Change. A change log by the change implementation (e.g., status of change); complementary comments
(7) Check-In, Check-Out Process: Once the change has been approved, the SCI has to be checked-out from the library for change; once the change has been implemented, the SCI must be checked-in to the library.
Implementing a change may take a long time.

The objective of Configuration Status Accounting is to answer questions such as:

- What is the status of a CR?
- What is the status of an approved CR?
  - Scheduled/not scheduled? Active? Completed?
- Who is in charge of implementing an approved CR?
- What is the average time to process a CR?
- What is the average efforts needed to process a CR?
- What is the number of CRs per SCI?
- Have all related SCIs been properly updated?
Configuration status accounting plays an important role in the success of large software projects where a large number of individuals are involved on the same project; the status accounting helps improving communication between those individuals involved.
Configuration Auditing

- Configuration Auditing Objective: to verify compliance with configuration control standards

- Configuration auditing is performed by auditors (external to the development team) who are in charge of determining if the defined processes are being followed and to ensure that the SCM goals are satisfied

- For example, it will determine whether software engineering an organizational standards (e.g., documentation standards, coding conventions) been properly followed?
Resources for Configuration Control

Traditional tools:

- make utility - operates on a makefile (includes a definition of the SCIs and their dependencies and a procedure for re-building the system if any of the components of the system has changed since the last built)
  Originally for the UNIX; migrated to the PC environments

- SCCS utility - administration programs for Source Code Control System (SCCS)

- RCS utility - a revision control that creates new revision control files or changes attributes of existing ones. An RCS file contains multiple revisions of text, an access list, a change log, descriptive text, and some control attributes.

- Many other similar tools, e.g., diff

For exclusive use of Professor Saiedian’s students
WWW Resources for Configuration Control

Comprehensive list of free and commercial products:
www.cmtoday.com/yp/configuration_management.html
Configuration Control/Request

Change Samples

- Standards
  - ANSI/IEEE Std 828 SCM Plans
  - ANSI/IEEE Std 1042 Guide to SCM
An Outline of a CM Plan

1. **Introduction:** Purpose, Scope, Definition, Acronyms

2. **Management:** organization of the project and baselines, SCM responsibilities, who does what

3. **CM activities:** how the organization will perform the CM activities (identification, control, status accounting, audit)

4. **Tools, Techniques and Methodologies:** technical details of implementing CM

5. **Supplier Control** (sub-contractor software; vendor software)

6. **Records Collection and Retention**