

# Software Configuration Management

# Change

---

- Changes will happen in all work products during software development and maintenance
- Change increases the level of confusion among software engineers who are working on a software project

## Confusion

- Confusion arises when changes are not analyzed before they are made
- Not Recorded before they are implemented
- Not Reported to those who need to know

*We need to minimize this confusion, or else our projects will get out of control*

---



# Sources of Change

---

- New business or market conditions dictate changes in product requirements or business rules
- Customer demand - modification of data produced by information systems, functionality delivered by products, or services delivered by computer-based system
- Changes in project priorities or software engineering team structure
- Budgetary or scheduling constraints cause a redefinition of the system or product



# Questions to be answered

---

- ▶ What is the correct version of the software module that I have to continue its coding?
  - ▶ Who can provide me with an accurate copy of last year's version 4.1 of the TMY software system?
  - ▶ What version of the design document matches the software version we are currently adapting to a new customer's requirements?
  - ▶ What version of the software system is installed at ABC Industries?
  - ▶ What changes have been introduced in the version installed at the ABC Industries' site?
  - ▶ Where can I find the full list of customers that use version 6.8 of our software?
- 



# SCM Introduction

---

- ▶ QA steps must be planned to provide accurate responses to the questions similar to the above discussed
- ▶ Software configuration management (SCM) is the SQA component assigned to manage changes and supply accurate answers to the discussed questions
- ▶ SCM deals with all the issues related to
  - ▶ Control of software changes
  - ▶ Proper documentation of changes
  - ▶ Registering and storing the approved software versions
  - ▶ Provision of the relevant information
  - ▶ Supply copies of registered versions throughout the software system's life cycle.



# Software configuration management – definition

---

*“Configuration Management (CM) is the process of identifying and defining the configuration items in a system, controlling the release and change of these items throughout the system lifecycle, recording and reporting the status of configuration items and change requests, and verifying the completeness and correctness of configuration items.”*



# SCM Plan

---

- ▶ SCM Plan-provides information on
  - ▶ Requirements and procedures necessary for the configuration management activities
- ▶ It specifies
  - ▶ Who will be responsible for accomplishing the planned activities
  - ▶ What activities will be performed
  - ▶ When the activities will be performed
  - ▶ What tools and human resources will be required



# Software configuration items /software configuration versions –definitions

---

## **Software configuration item (SCI) or configuration item (CI)**

- ▶ A unit of software code, a document is defined as an SCI if it is assumed that it may be needed for further development of the software system and/or its maintenance.
- ▶ A software configuration is composed of as many SCIs as the developers assume will be needed in the future, with each SCI approved, identified and registered.





# Software configuration items /software configuration versions –definitions

---

## **SCI version**

The approved state of an SCI at any given point of time during the development or maintenance process.

## **Software configuration version**

An approved selected set of documented SCI versions that constitute a software system, where the activities to be performed are controlled by software configuration management procedures.



# Configuration Identification

---

- ▶ Configuration identification consists of selecting the configuration items
- ▶ Recording their characteristics in technical documentation such as specifications, drawings..
- ▶ An important aspect of configuration identification is the use of a formal naming convention for each entity.
- ▶ An SCI is identified by its name in combination with its version and revision numbers.



# Software Configuration Item

---

The following are examples of items managed in the SCM system.

- Management plans
- Specifications (requirements, design)
- User documentation
- Test plans, test design, case, and procedure specifications
- Test data
- Source code
- Executable code
- Databases
- Maintenance documentation



# Common types of software configuration items

---

## ***Documents***

- ▶ Software development plan (SDP)
- ▶ Software requirements document (SRD)
- ▶ Preliminary design document (PDD)
- ▶ Critical design document (CDD)
- ▶ Database description
- ▶ Software test plan (STP)
- ▶ Software test procedure (STPR)
- ▶ Software user manuals
- ▶ Software maintenance manuals



# Numeration conventions for identification of SCI

---

- ▶ Numeration conventions have been formulated to identify SCIs
- ▶ The most commonly used is decimal numeration
- ▶ which indicates the successive version and revision numbers and is registered accordingly.

## **For example**

- ▶ An SCI design document captioned DD-7 may have several versions and revisions identified as DD-7 Ver.1.0, DD-7 Ver.1.1, DD-7 Ver.2.0, DD-7 Ver.3.0, DD-7 Ver.3.1, DD-7 Ver.3.2, etc.,
- ▶ where the first number represents the version and the second the revision.
- ▶ An SCI is identified by its name in combination with its version and revision numbers.



Table 18.1: Pepper Mountain Travel (PMT) – software configuration versions, including SCI versions		
SCI version	Release and release date	
	PMT Version 6.0 January 6, 2002 SCI version in the release	PMT Version 7.0 January 22, 2003 SCI version in the release
SRD	Ver. 1	Ver. 1
CDD	Ver. 3	Ver. 4
STP	Ver. 3	Ver. 4
SIP	Ver. 2	Ver. 2
VDD	Ver. 6	Ver. 7
Code Module 1	Ver. 3	Ver. 5
Code Module 2	Ver. 8	Ver. 8
Code Module 3	Ver. 2	Ver. 2
Test cases file	Ver. 3	Ver. 4
CL compiler	Ver. 5	Ver. 7
Software user manual	Ver. 6	Ver. 7
▶		

# Baselines

---

A baseline is

- ▶ A specification or product that has been formally reviewed and agreed upon
  - ▶ That serves as the basis for further development
  - ▶ Can only be changed through formal change control procedures.
- 
- ▶ During each phase of the software lifecycle, configuration items that are completed and accepted by the approval authority become the baseline for that phase.



# Baselines

---

Typical baselines include:

- ▶ The statement of system requirements (functional baseline)
- ▶ High level design (allocated baseline)
- ▶ Detailed design (design baseline)
- ▶ The software product at the completion of system test (product baseline)
- ▶ The software product in its operational environment (operational baseline).





**Requirements**

*Functional Baseline*

**Preliminary  
Design**

*Allocated Baseline*

**Detailed  
Design**

*Design Baseline*

**Build**

**Test**

*Product Baseline*

**Deploy**

*Operational Baseline*

**Maintain**

# Configuration Control (Change Control)

---

- ▶ Configuration control is a critically important feature of configuration management.
- ▶ Configuration control ensures that consistency between component parts of a system is maintained.
- ▶ It reduces the possibility of making changes that may later adversely affect functionality.



# Configuration Control (Change Control)

---

The configuration control process involves three elements:

- Procedures for controlling changes to a software product.
- Levels of authority (e.g., Configuration Control Board) for formally evaluating and approving or disapproving proposed changes to the software.
- Documentation



# Configuration Control (Change Control)

---

The main tasks of software change control can be described as:

- ▶ Examining change requests and approving implementation those requests that Qualify
- ▶ Controlling the changes and assuring the quality of approved changes.
- ▶ Documenting the approved changes.
- ▶ Applying mechanisms that prevent more than one team from simultaneously introducing changes into the same SCI.



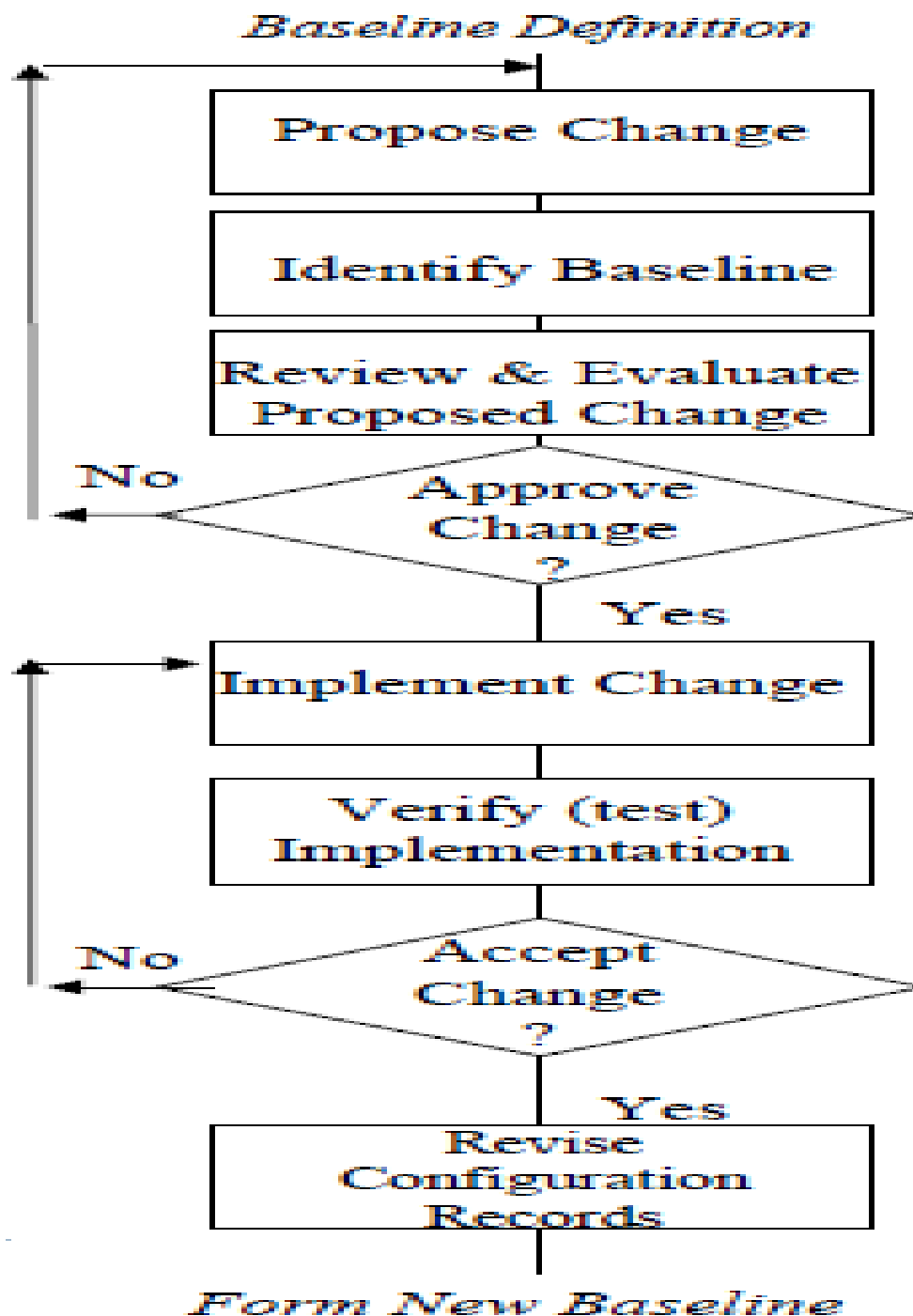
# Procedures for Controlling Changes

---

Changes occur at all phases in the software lifecycle.

- ▶ Design or implementation changes may be necessary if requirements change, or when deficiencies are identified in the design or implementation approach
- ▶ Testing may uncover defects that require changes in the code , design and requirements.
- ▶ Changes must be made to the right version of the code
- ▶ Testing must verify performance of the change and the integrity of the remaining software
- ▶ All associated documentation must be updated





# Levels of Authority

---

## **Configuration Control Boards**

- ▶ Also called
  - ▶ Software change control authority (SCCA)
  - ▶ Software change control board (SCCB)
  - ▶ Change control authority (CCA)
- ▶ Mostly composed of senior level managers



# Levels of Authority

---

Some of the functions of the CCB include:

- ▶ Directing the entire configuration management effort
- ▶ Resolving all problems and situations that arise during the effort
- ▶ Determining which products should be baseline or managed, the method to be used, and the order in which they should be done
- ▶ Changes are implemented and tested according to standard procedures
- ▶ Design or requirements specifications are not violated
- ▶ Software changes are implemented according to cost and schedule constraints





# Benefits of SCM

---

The effective use of an SCM system also:

- Permits the orderly development of software configuration items.
- Ensures the orderly release and implementation of new or revised software products.
- Ensures that only approved changes to both new and existing software products are implemented.
- Ensures that the software changes that are implemented are in accordance with approved specifications.
- Ensures that the documentation accurately reflects updates.
- Evaluates and communicates the impact of changes.
- Prevents unauthorized changes from being made.

