

Software Configuration Management

Lecturer: Ngo Huy Bien
Software Engineering Department
Faculty of Information Technology
VNUHCM - University of Science
Ho Chi Minh City, Vietnam
nhbien@fit.hcmus.edu.vn

Objectives

- To identify *configuration items*
- To *control* configuration
- To report *configuration status*
- To perform *configuration audit* and review
- To control *configuration changes*
- To create a *configuration management plan*



References

1. Jessica Keyes, *Software Configuration Management*, 2004.
2. Anne Mette Jonassen Hass, *Configuration Management Principles and Practice*, 2002.
3. Roger S. Pressman. *Software Engineering -- A Practitioner's Approach*. 5th Edition. McGraw-Hill. 2001.
4. Susan A. Dart. *The Past, Present, and Future of Configuration Management*. 1992.



Development Requirements



Source Code Revert

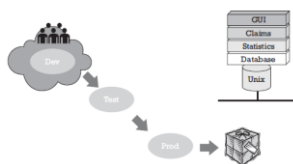
Source Code Sharing



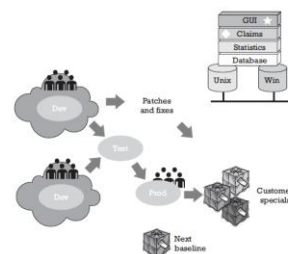
Source Code Branching



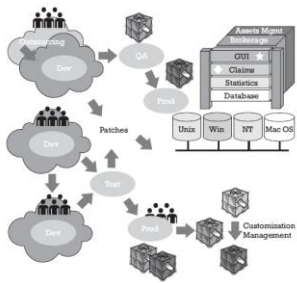
First Year



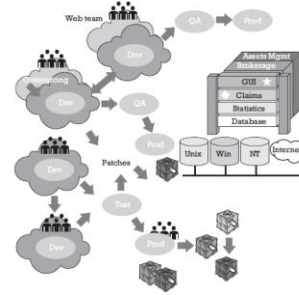
Second Year



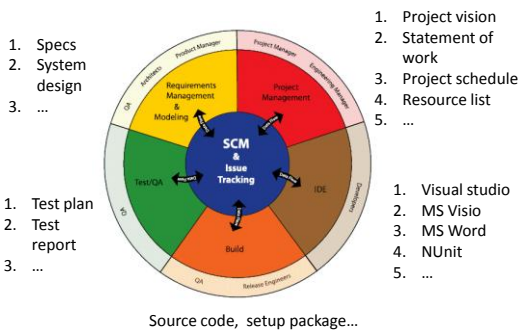
Third Year



Fourth Year



Software Artifacts



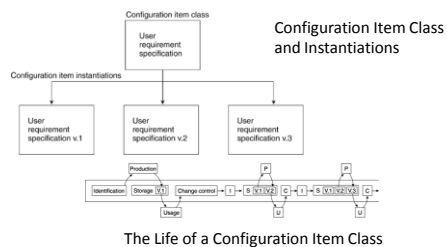
What is CM? [1]

Software configuration management (SCM, or just plain CM) is an organizational framework — that is, a discipline — for managing the evolution of computer systems throughout all stages of systems development.



Configuration Item [2]

A *configuration item* is any possible part of the development or delivery of a system or product that it's necessary to identify, produce, store, use, and change individually.



Electronic Objects



It must be stored electronically.

An unlimited number of copies may be easily produced.

Copies may be in either *electronic* or *physical* form.

Physical Objects



It cannot be stored on a computer but must be stored in a *physical place*.

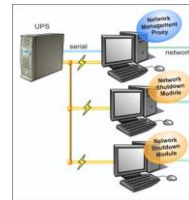


When a copy of the object is delivered, the number of available copies is reduced.



Most traditional configuration management tools cannot handle a physical object without a *proxy-object* in electronic form.

Objects Of A Product

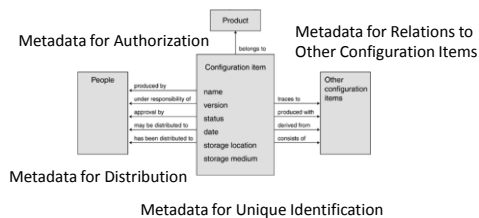


Hardware – Network



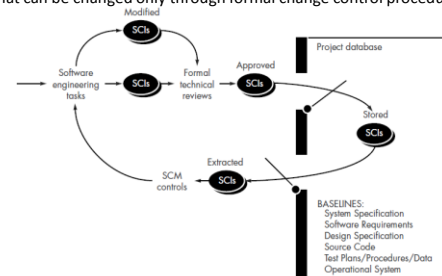
Procedure descriptions, service agreements or contracts, training material (computer-assisted training material, teacher's manual, slides) or user manuals.

Metadata For A Configuration Item



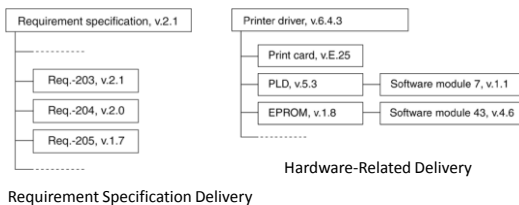
Baseline [3]

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 procedures.



Deliveries [2]

Deliveries are hierarchies of configuration items and may be constructed of other deliveries in a ramified hierarchy.



Requirement Specification Delivery

Milestone

A *milestone* is a specific point with a specified outcome.

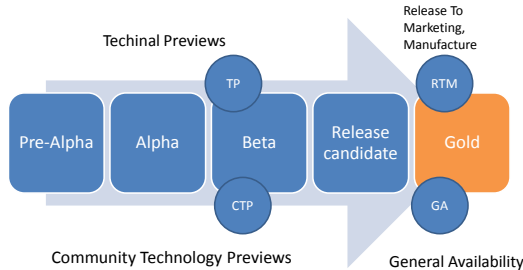


Included CIs	01/10/2011	03/15/2011	04/30/2011	06/30/2011
Project plan	1.0	3.0	4.1	4.1
Requirement specification	1.0	1.3	1.6	1.6
Architectural design	—	1.2	1.3	1.3
Detailed design	—	1.0	1.2	1.3
User manual	—	—	1.1	1.1
Complete system	—	—	1.0	1.1
Release note	—	—	—	1.0

A dash indicates that the configuration item is not part of the delivery.

Release

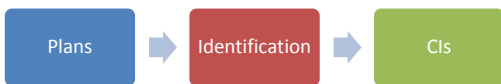
A **software release** is the distribution of an initial or upgraded version of a computer software product. For example, executable program, documentation, release notes, and configuration data.



Configuration Management Activities



Identifying Configuration Items



1. Products delivered to the customer
2. Designated internal work products
3. Acquired products
4. Tools and other capital assets of the project's work environment
5. Other items used in creating and describing these work products

Hardware
Software
Product Source Code
Documents

Assign Unique Identifiers To CIs

`<SYSTEM>[<TLA>]_[<SUBSYSTEM>]_[<TLA>]_[R|A|B|X][<Y>][<Z>].[BL<#>]`

PLN	Project Plans	R A B	Stand for release, alpha, or beta
SOW	State Of Work		
USC	Use Cases	<X>	Integer, stands for a major release (e.g. 1)
SRD	Software Requirements Document	<Y>	Integer (optional), stands for a minor release
SDD	Software Design Document	<Z>	Integer (optional), stands for an alternative release (patches, ports, etc.)
SRC	Source Code Files	BL	Stands for base level (an internal release)
TSP	Test Plan	#	Integer, for internal releases

Windows_PLN_B1.0
Windows_IE_USC_R3.5
Windows_R7.0

Source Code?

Specifying The important Characteristics Of CIs



author



file type



programming language

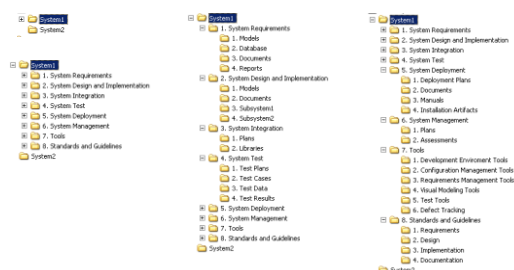


when to control

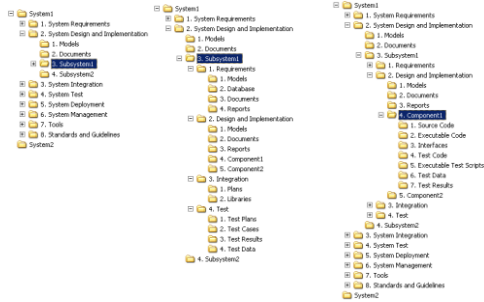


responsible people

Developing System Directory Structure



Developing Subsystem And Component Directory Structure



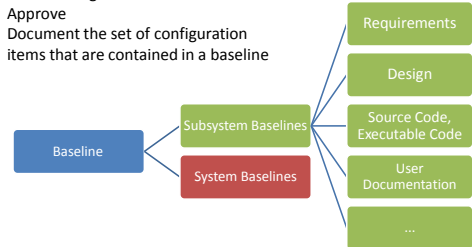
Identifying Baselines

Activities

1. Identify role and responsibility of people creating and approving baseline
2. Select Configuration Items
3. Approve
4. Document the set of configuration items that are contained in a baseline

Outputs

1. Baselines
2. Description of baselines



Define Configuration Management System



Storage media



Procedures

- Ensure that a configuration item will not **disappear** or be damaged
- It can **be found** at any time and **delivered** in the condition in which you expect to find it.
- Record is kept to indicate **who** has been given the item or a copy of it.

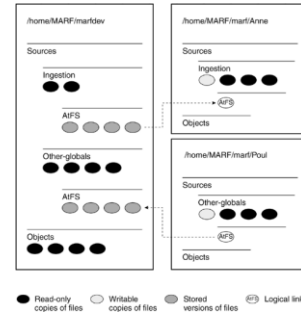


TortoiseSVN

The tools for accessing the configuration system

Defining Library

The Controlled Library



The Dynamic Library

The Dynamic Library

Defining Item Approval

Unit Approval Form	
Unit Name	
Version (v.x)	
Developer	
Approver	
Unit Approval Cycle	
Unit ready for Test (Shipment/Release)	
Conditions	All code elements covered by unit test if No - state coverage and reason:
Remarks	Why not has been performed after new build
Developer's Signature	Date
Unit ready for Unit Test	
Conditions	The guidelines in the Test Plan are followed A dry run of the test has been performed after new build Flow messages for performing the unit test are available and under CDD (or development) + release Unit ready for coverage of test items 100% (i.e. the build of the test is not responsible for the results when executing it) Statements not covered by the test have been verified by inspection
Remarks	
Approver's Signature	Date

An item approval will typically be a form; either **paper based** or **electronic**.

Item approval is evidence that items satisfy the criteria for placing under configuration management.

Content

1. Configuration item concerned
2. Dated signatures (electronic or other) by the producer, the person responsible, and the approver
3. Condition(s) for approval
4. Related metadata

Defining Change Request

A **change request** is a document containing a call for an adjustment of a system.

1. A **wrong formulation**, caught during the review of a document
2. A **coding mistake**
3. An **enhancement**
4. A mistake found in the integration **test**
5. An inquiry to a helpdesk about a problem in connection with **usage** of a system
6. A change required in the code because of an upgrade to a new version of the middleware supporting the system, which may not be **backward compatible**

Software Change Request (SCR) Form	
CHANGE REQUEST INITIATION	Integration
Date Submitted	System Name
Version Number	Version Number
CONFIGURATION ITEM	Software
CHANGE TYPE	New Development
REASON	Improvement
PRIORITY	Urgent
CHANGE DESCRIPTION	Describe functional and/or technical information. Use attachment if necessary.

Content

1. The change: identification, identification of the underlying event, configuration item concerned, and priority
2. Phase information for the change: phase, date and time, name of the person responsible, description

Defining Release Request

Release Request Form	
Configuration Item Name	
Version	Single / Multiple
Current Version of Configuration	
Requester	
Approval of Requester(s)	
Remarks	
Release Requester's Signature	Date
Release Information	
Release	Release approved if No - Still under review
Delivery	Release form created
Approval	Completed for control
Delivery medium	Complete delivery controlled
Approval	
Delivery medium	
Approval	
Delivery medium	

A release request may be a form, either **paper based** or **electronic**.

A **release request** is a document defining what is released from the configuration management library, and to whom.

Content

1. Configuration item concerned
2. Dated signature(s) by requester or requesters
3. If desired, dated signature by the person responsible
4. Reason for release request
5. Delivery medium and/or destination

Establishing A Configuration Management System

Activities

1. Establish a mechanism to manage **multiple control levels** of configuration management.
2. **Store and retrieve configuration items** in a configuration management system.
3. **Share and transfer configuration items** between control levels within the configuration management system.
4. **Store, update, and retrieve** configuration management **records**.



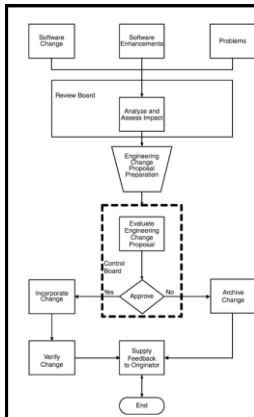
Outputs: Configuration Management Records

1. Revision history of configuration items
2. Change log
3. Copy of the change requests
4. Status of configuration items
5. Differences between baselines

Change Control

The **goal of configuration change control** is to establish mechanisms that will help ensure the production of quality software as well as ensure that each version of the software contains all necessary elements, and that all elements in a version will work correctly together.

1. Defining the change process
2. Establishing change control policies and procedures
3. Maintaining baselines
4. Processing changes
5. Developing change report forms



Status Reporting

Status reporting makes available, in a useful and readable way, the information necessary to effectively manage a product's development and maintenance.



Why status reporting?

1. Get information on change decisions
2. Assist future planning efforts
3. Review the complete configuration of a product or any of its component parts
4. Review maintenance information
5. Review documentation
6. Review source code

Activities

1. List baselines
2. Highlight current CIs and changed CIs
3. List changes

Configuration Audits

Configuration audits confirm that the resulting baselines and documentation conform to a specified standard or requirement.

Functional Configuration Audits (FCA) – Audits conducted to verify that the actual performance of the CI meets the requirements stated in its performance specification and to certify that the CI has met those requirements.



Physical Configuration Audit (PCA) – Audits conducted to verify that the related design documentation matches the design of the deliverable CI.

Performing Configuration Audits

1. Assess the integrity of the **baselines**.
2. Confirm that the configuration management **records** correctly identify the configuration items.
3. Review the structure **and integrity of the items** in the configuration management system.
4. Confirm the **completeness and correctness of the items** in the configuration management system. Completeness and correctness of the content is based on the requirements as stated in the plan and the disposition of approved change requests.
5. Confirm **compliance** with applicable configuration management standards and procedures.
6. Track **action items** from the audit to closure.

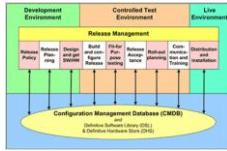


Outputs:

1. Configuration audit results
2. Action items

Release Management

Software release management contains identification, packaging, and delivery of the elements of a product.



1. Identify baselines
2. Create Configuration Status Accounting Reports (CSAR)
3. Perform PCA, FCA
4. Package files and documents
5. Receive client's confirmation

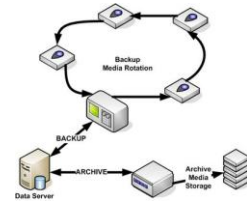
Backup And Archive

Benefits:

1. Recovery data when there are problems.
2. Support version control.

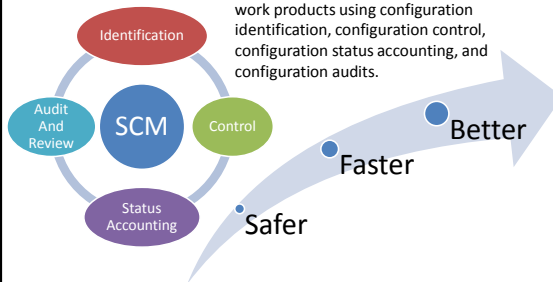
When project is finished:

1. Archive data of project
2. Archive or remove paper documents
3. Clean all information of project



Why CM?

The purpose of configuration management is to establish and maintain the *integrity* of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.



Configuration Management Roles



Configuration Manager

The person responsible for configuration management *implements*, maintains, and improves configuration management within the framework provided by management.



1. Transforming the company's *needs* and *requirements* for configuration management to relevant, practical procedures, resources, and tools
2. Selecting and testing configuration management *tools*
3. Updating information about *new versions* of existing tools and new tools
4. Following up on the *performance and efficiency* of configuration management
5. Making *status reports* to management with data analysis and recommendations for improvement

Librarian

1. Establishing the configuration management *library*—a controlled master library for storing configuration items
2. Maintaining and controlling the *contents* of the library
3. *Communicating* contents of the configuration management library
4. *Controlling* the configuration management library



Configuration Control Board

Configuration Manager

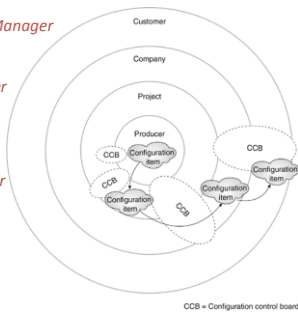
Librarian

Project Manager

Technical Lead

Test Lead

Quality Engineer



Related People

1. Analyst
2. Architect
3. Developer
4. Integrator
5. Tester
6. Customer Contact
7. People Being Responsible For Assets Operation, Process Management Support

Analyst

1. Identifying *relevant configuration items*
2. Placing relevant items in *storage* after due approval
3. Producing appropriate *event registrations* for the items used in connection with analysis (such as contracts or user requirement specifications)



1. Extracting *related configuration items* as the basis for producing analysis objects, such as contracts or user requirement specifications
2. Getting information about the *status* and history of these items
3. Getting *trace analysis* results toward these items, to ensure the analysis covers all requirements

Architect

1. Identifying relevant configuration items (*design documents*)
2. Placing relevant configuration items in *storage* after due approval
3. Producing appropriate *event registrations* for items used in connection with design work, such as user requirement specifications and software requirements specifications



1. Extracting *related configuration items* as the basis for producing design items, such as software requirements specifications
2. Getting information on the *status* and history of these items
3. Getting *trace analysis* results toward these items, to ensure that the design covers all demands

Developer

1. Identifying relevant configuration items (*source code* and object files)
2. Placing relevant configuration items in *storage* after due approval
3. Producing appropriate *event registrations* for items used in connection with programming, such as requirement specifications or design



1. Extracting related configuration items as the basis for producing programming objects, such as software requirement specifications and design
2. Getting information on the status and history of these items
3. Getting trace analysis results toward these items, to ensure that the code and related objects cover at least the design and possibly also explicitly all software requirements

Integrator

1. Identifying *relevant configuration items* (build scripts and, not least, deliveries in the form of larger and larger subsystems)
2. Placing relevant items in *storage* after due approval
3. Producing appropriate *event registrations* for items used in connection with integration, such as source code



1. Extracting *related configuration items* as a basis for integration, such as architectural design, development plans, and test plans
2. Extracting *configuration items* from which their own items must be produced
3. Getting information on the *status and history* of these items

Tester

1. Identifying *relevant configuration items* (test plans, descriptions, scripts, and data, and releases for an entire test, including test environment)
2. Placing relevant items in *storage* after appropriate approval
3. Producing appropriate *event registrations* for items used in connection with testing, such as source code or (sub)systems



1. Extracting *related configuration items* as the basis for testing, such as individual configuration items or, more important, deliveries in terms of integrated (sub)systems
2. Getting information on the *status and history* of these items
3. Getting information about relevant event registrations and their *progress*

Project Manager

1. Producing and updating a configuration management **plan** in agreement with the overall project plan
2. Identifying necessary configuration management **roles** for the project
3. Assigning **responsibility** for configuration management activities in accordance with identified roles
4. Allocating **resources** for configuration management
5. Following up on planned configuration management **activities**



1. Status reports from the configuration management system concerning configuration items
2. Information about event registrations and their progress
3. Measurements produced from the configuration management system, concerning both configuration management itself and other processes

Customer And Customer Contact



1. Participating in one or more configuration control boards
2. Creating event registrations
3. Approving produced configuration items

1. Producing documentation to fulfill the customer's configuration management requirements
2. Receiving and possibly performing quality assurance on deliveries from the customer, such as user requirement specifications, and possibly performing internal configuration management on them
3. Possibly forwarding event registrations to the customer
4. Receiving event registrations from the customer, such as in connection with a review of documentation or with early user or acceptance tests



Subcontractor And Subcontractor Contact



Study, understand, and use the contractor's configuration management system

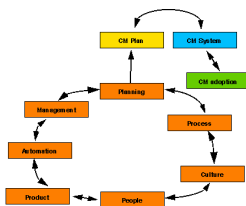
Deliver event registrations or change requests (depending on the form of cooperation) to the subcontractor



Configuration Management Plan



Configuration Management Plan



CM plan is the actual plan that will be implemented to address the CM needs. It gives all the procedures, policies, schedules, responsibilities, etc.

A plan describes what you will do and a procedure describes how it will be done.

✓The plan documents the CM process and as such acts as the tool used to gain project and management support for the process.

- ✓The plan forces you to define and describe the process.
- ✓The plan causes you to think about what you will do and how you will do it.
- ✓The plan serves as a contract vehicle for the project.

Interviews



Do standards aid in the development of a CM plan?



Should CM procedures be part of the CM plan or be separate?



Is the CM plan updated throughout the project life cycle?



Was the CM plan used after it was developed? If so, by whom and how?

Interviews (cont.)



Is there a need for a CM plan at the company/division level as well as at the project level?



Are there significant differences between a CM plan written for a development project and a CM plan written for a maintenance project?



Are there significant differences between a CM plan written for hardware versus software?

Interviews (cont.)



Are there significant differences between a CM plan written for a large project versus a small project?

What makes a CM plan hard to write?

1. How to perform CM?
2. What processes you will implement?
3. Lack of a defined CM process makes it impossible to write the CM plan



Would having an automated tool to assist in developing a CM plan help?

Outline of a Model CM Plan

1.0 INTRODUCTION

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions
- 1.4 References
- 1.5 Tailoring

2.0 SOFTWARE CONFIGURATION MANAGEMENT

- 2.1 SCM organization
- 2.2 SCM responsibilities
- 2.3 Relationship of CM to the software process life cycle
 - 2.3.1 Interfaces to other organizations on the project
 - 2.3.2 Other project organizations CM responsibilities



Outline of a Model CM Plan (cont.)

3.0 SOFTWARE CONFIGURATION MANAGEMENT ACTIVITIES

3.1 Configuration Identification

3.1.1 Specification Identification

- Labeling and numbering scheme for documents and files
- How identification between documents and files relate
- Description of identification tracking scheme
- When a document/file identification number enters controlled status
- How the identification scheme addresses versions and releases
- How the identification scheme addresses hardware, application software system software, COTS products, support software (e.g., test data and files), etc.

3.1.2 Change Control Form Identification

- Numbering scheme for each of the forms used



Outline of a Model CM Plan (cont.)

3.1.3 Project Baselines

- Identify various baselines for the project
- For each baseline created provide the following information:
 - How and when it is created
 - Who authorizes and who verifies it
 - The purpose
 - What goes into it (software and documentation)

3.1.4 Library

- Identification and control mechanisms used
- Number of libraries and the types
- Backup and disaster plans and procedures
- Recovery process for any type of loss
- Retention policies and procedures
 - What needs to be retained, for who, and for how long
 - How is the information retained (on-line, off-line, media type and format)



Outline of a Model CM Plan (cont.)

3.2 Configuration Control

3.2.1 Procedures for changing baselines (procedures may vary with each baseline)

3.2.2 Procedures for processing change requests and approvals-change

classification scheme

o Change reporting documentation

o Change control flow diagram

3.2.3 Organizations assigned responsibilities for change control

3.2.4 Change Control Boards (CCBs) - describe and provide

the following information for each:

- o Charter
- o Members
- o Role
- o Procedures
- o Approval mechanisms

3.2.5 Interfaces, overall hierarchy, and

the responsibility for communication between multiple CCBs, when applicable

3.2.6 Level of control - identify how it will change throughout the life cycle, when applicable

3.2.7 Document revisions - how they will be handled

3.2.8 Automated tools used to perform change control



Outline of a Model CM Plan (cont.)

3.3 Configuration Status Accounting

3.3.1 Storage, handling and release of project media

3.3.2 Types of information needed to be reported and the control over this information that is needed

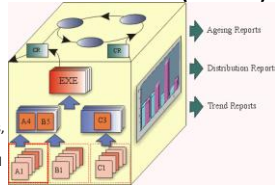
3.3.3 Reports to be produced

(e.g., management reports, QA reports, CCB reports) and who the audience is for each and the information needed to produce each report

3.3.4 Release process, to include the following information:

- o What is in the release
- o Who the release is being provided to and when
- o The media the release is on
- o Any known problems in the release
- o Any known fixes in the release
- o Installation instructions

3.3.5 Document status accounting and change management status accounting that needs to occur



Outline of a Model CM Plan (cont.)

3.4 Configuration Auditing

3.4.1 Number of audits to be done and when they will be done (internal audits as well as configuration audits); for each audit provide the following:

- o Which baseline it is tied to, if applicable
- o Who performs the audit
- o What is audited
- o What is the CM role in the audit, and what are the roles of other organizations in the audit
- o How formal is the audit

3.4.3 All reviews that CM supports; for each provide the following:

- o The materials to be reviewed
- o CM responsibility in the review and the responsibilities of other organizations



Outline of a Model CM Plan (cont.)

4.0 CM MILESTONES

- Define all CM project milestones (e.g., baselines, reviews, audits)
- Describe how the CM milestones tie into the software development process
- Identify what the criteria are for reaching each milestone



5.0 TRAINING

- Identify the kinds and amounts of training (e.g., orientation, tools)

6.0 SUBCONTRACTOR/VENDOR SUPPORT

- Describe any subcontractor and/or vendor support and interfacing, if applicable

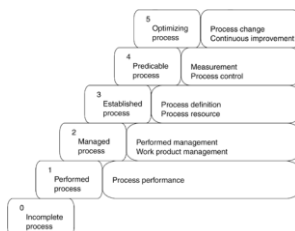


Configuration Management Solution



ISO 15504 (SPICE) [4]

Software Process Improvement and Capability Determination



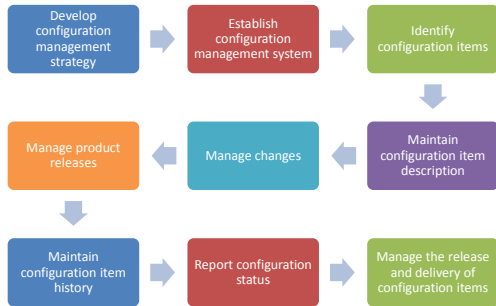
Work product management means that for any given process area to obtain level 2, all **relevant work products** from the performance of the process area must be placed under configuration management.

Goals

- A configuration management **strategy** will be developed.
- All items generated by the process or project will be identified, defined, and **base-lined**.
- **Modifications** and releases of the items will be controlled.
- The status of the items and modification requests will be recorded and **reported**.
- The **completeness** and consistency of the items will be ensured.
- Storage, handling, and **delivery** of the items will be controlled.

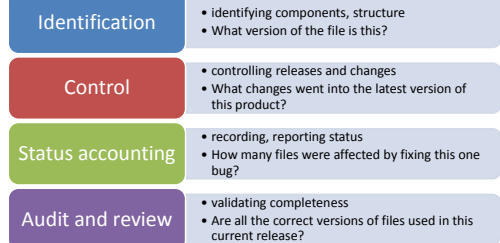


Practices



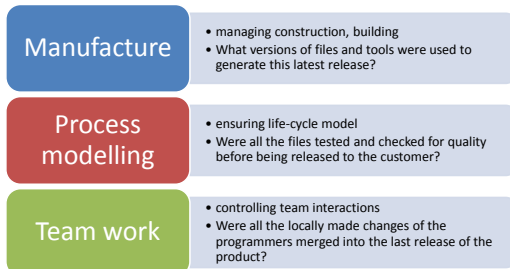
Definition of CM

Standard definition (IEEE 729-1983)

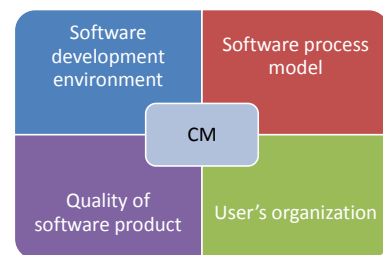


Definition of CM

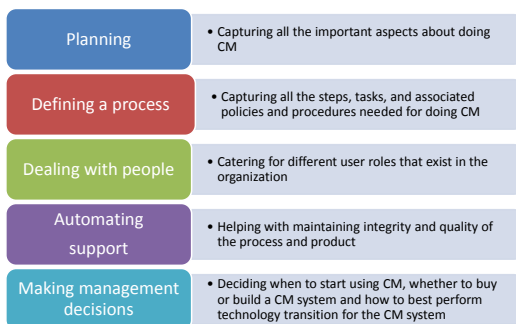
Based on existing CM systems, broaden definition



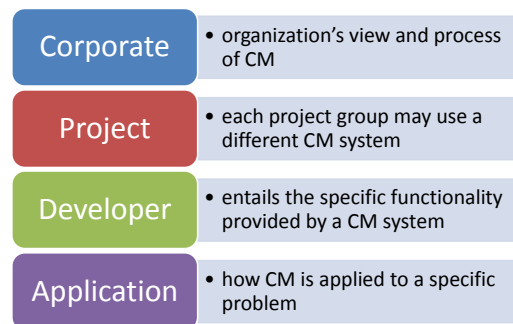
Effects



CM Solution



Perspectives



In-house

Manual procedures
and policiesCompiling
codeTracking
and
dealing
with bugsVersion
controlThird-
party
solutions?

CM Technology

CM concepts

Management

Political

Process
orientation

Standardization

CM Services Model

The Power of
Information

Workflow

Automation

A Branch
too Far

Thank You For Your Time

