

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



Contents

- I. Quick start
- II. Configuration management terminologies
- III. Configuration management activities
- IV. Configuration management roles
- V. Configuration management plan
- VI. Configuration management solution



References

1. Jessica Keyes (2004). Software Configuration Management.
2. Anne Mette Jonassen Hass (2002). Configuration Management Principles and Practice.
3. Roger S. Pressman (2010). Software Engineering : A Practitioner's Approach.
4. Susan A. Dart (1992). The Past, Present, and Future of Configuration Management.



Quick Start: Collaboration

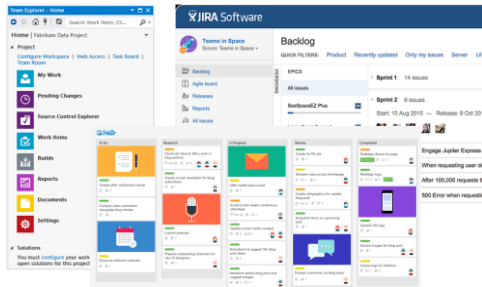


Hands-On Collaboration

1. Zoom. <https://zoom.us/>
2. Skype. <https://www.skype.com/en/>
3. Discord. <https://discordapp.com>



Quick Start: Tracking Tools



Hands-On Tracking Tools

1. Trello. <https://trello.com/>
2. Asana. <https://asana.com>
3. Atlassian Cloud. <https://www.atlassian.com/software/free>



Quick Start: Source Control



Source Code Sharing



Source Code Branching and Merging



Source Code Reversion

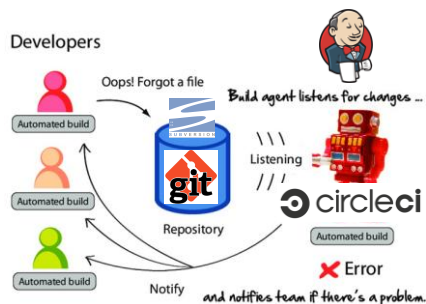


Hands-On Source Control

1. GitHub. <https://github.com/>
2. GitLab. <https://about.gitlab.com/>
3. Git. <https://git-scm.com/>
4. Apache Subversion. <https://subversion.apache.org/>
5. VisualSVN Server. <https://www.visualsvn.com/server/>
6. TortoiseSVN. <https://tortoisesvn.net/>



Quick Start: Build Automation



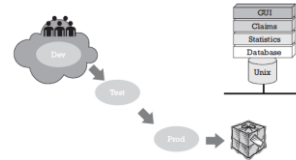
Hands-On Build Automation

1. GitLab. <https://about.gitlab.com/>
2. CircleCI. <https://circleci.com/>
3. Wercker. <https://devcenter.wercker.com/>
4. Jenkins. <https://jenkins.io/doc/tutorials/>
5. buddybuild (for iOS). <https://buddybuild.com/>

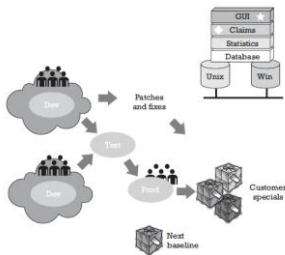




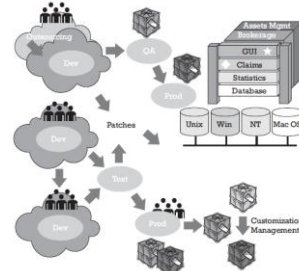
First Year



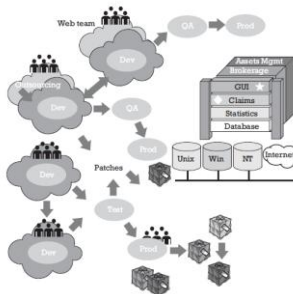
Second Year



Third Year



Fourth Year

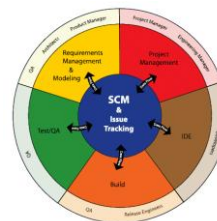


Software systems are *very complex*.

Creating a Release

1. Specs
2. System design

1. Test plan
2. Test report



Source code, setup package

1. Project vision
2. Statement of work
3. Project schedule
4. Resource list

1. Visual studio
2. MS Visio
3. MS Word
4. NUnit



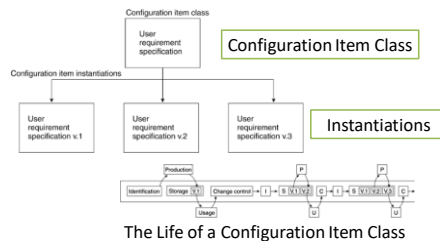
What is CM? [1]

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. *Examples?*



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

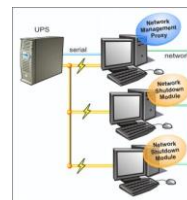


It cannot be stored on a computer but must be stored in a *physical place*.

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

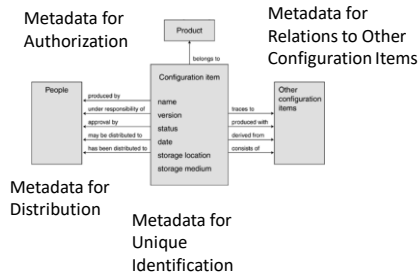


Hardware – Network



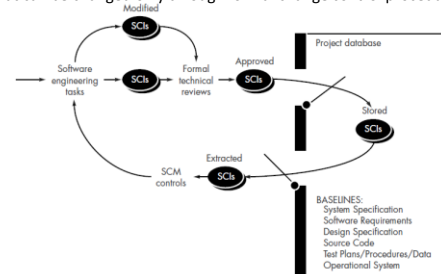
Procedure descriptions, service agreements or contracts, training material (computer-assisted training material, teacher's manual, slides) or user manuals.

Configuration Item Metadata [2]



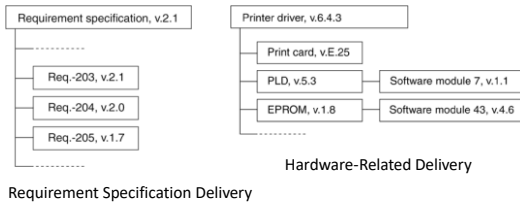
Baseline [3]

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.



Milestone

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

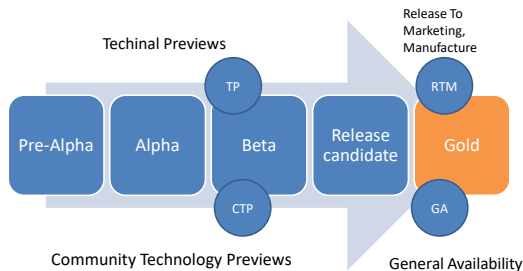


Included CIs	01/10/2011	03/15/2011	04/30/2011	06/30/2011
Project plan	1.0	3.0	4.1	4.1
Requirement specification	1.0	1.3	1.6	1.6
Architectural design	—	1.2	1.3	1.3
Detailed design	—	1.0	1.2	1.3
User manual	—	—	1.1	1.1
Complete system	—	—	1.0	1.1
Release note	—	—	—	1.0

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

Release

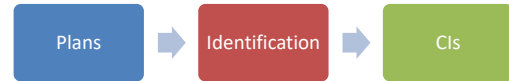
A **software release** is the distribution of an initial or upgraded version of a computer software product. For example, executable program, documentation, release notes, and configuration data.



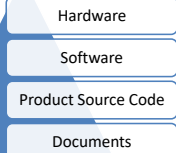
Configuration Management Activities



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



Assigning 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		
TSP	Test Plan	BL	Stands for base level (an internal release)
		#	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

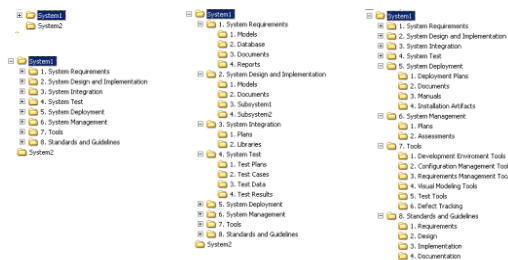


when to control



responsible people

Developing System Directory Structure



Developing Subsystem And Component Directory Structure



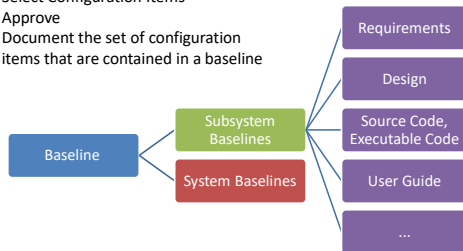
Identifying Baselines

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



Defining Item Approval [2]

Unit Approval Form	
Unit Name	
Version	
Configuration	
Approval	
Unit Approval Cycle	
Unit ready for Test Readiness Review	
Conditions	All code components covered by unit test. If No - state coverage and reason. Dry run has been performed after new build.
Remarks	
Reviewer'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. If No - state coverage and reason. If No - state coverage and reason. Valid reason for coverage of less than 100%. If No - the result of the test is not comparable to the code subject to testing. Coverage not covered by the test has been verified by inspection.
Remarks	
Reviewer'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	
SCR #	
CHANGE REQUEST INITIATION	Date Submitted: _____ System Name: _____ Version Number: _____ CONFIGURATION ITEM: _____ CHANGE TYPE: New Requirement, Requirement Change, Design Change, Other REASON: Bug, Performance, Functionality, Testing, Defect PRIORITY: Critical, High, Medium, Low CHANGE DESCRIPTION: (Brief functional order level description; use attachment if necessary)
Attachments: The file	

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	
Type	Single / Delivery
Delivery Medium or Destination	
Requester	
Reason(s)	
Remarks	
Non-Requester's Signature	Date
Release Information	
Release	Release approved / If No - state reason
Delivery	Release Note created / Components for install / Complete delivery controlled
Remarks	
Reviewer's Signature	Date

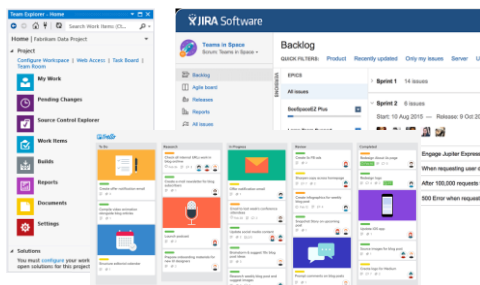
A release request may be a form, either *paper based* or *electronic*.

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 Collaboration Environment



Establishing 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.



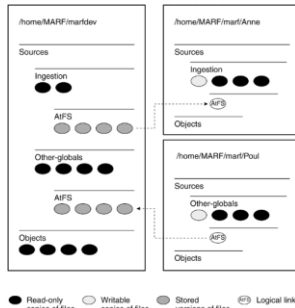
git



The tools for accessing the configuration system

Defining Libraries

The
Controlled
Library



The
Dynamic
Library

The
Dynamic
Library

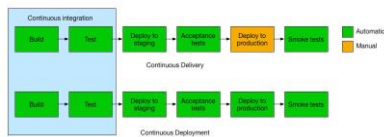
Operating Configuration Management System

- Activities:
 - Establish a mechanism to *manage multiple control levels* of configuration management.
 - Store and retrieve* configuration items in a configuration management system.
 - Share and transfer* configuration items between control levels within the configuration management system.
 - Store, update, and retrieve* configuration management records.
- Outputs: Configuration Management *Records*
 - Revision history of configuration items
 - Change log
 - Copy of the change requests
 - Status of configuration items
 - Differences between baselines



Establishing CI/CD Process

- <https://www.atlassian.com/continuous-delivery/ci-vs-cd>
- Continuous integration:** The developer's changes are validated by creating a build and running automated tests against the build.
- Continuous delivery** is an extension of continuous integration: on top of having automated your testing, you also have automated your release process and you can deploy your application at any point of time by clicking on a button.
- Continuous deployment:** Every change that passes all stages of your production pipeline is released to your customers. There's no human intervention, and only a failed test will prevent a new change to be deployed to production.



Implementing DevOps

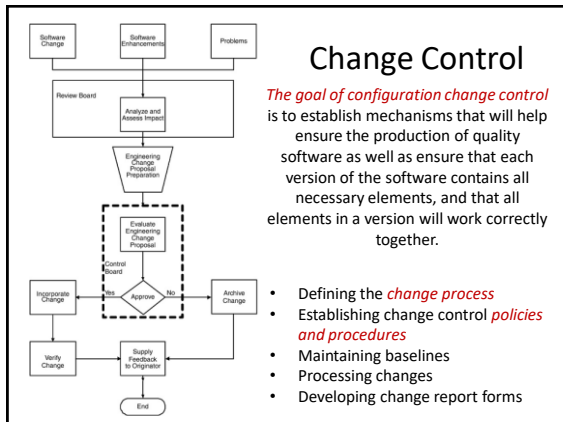
- Len Bass, Ingo Weber and Liming Zhu (2015). DevOps: A Software Architect's Perspective.
- Gene Kim, Jez Humble, Patrick Debois, and John Willis (2016). The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations.



Developers' Tasks

- Create configuration setting for a feature.
- Use IF ELSE for enabling or disabling code of the feature.





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?

- Get information on change decisions
- Assist future planning efforts
- Review the complete configuration of a product or any of its component parts
- Review maintenance information
- Review documentation
- Review source code

Activities:

- List baselines
- Highlight current CIs and changed CIs
- List changes

Configuration Audits

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



Functional Configuration Audits

- *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.
- *Why FCA?*



- A multi million dollar system is shipped to a customer with three important features *missing*.
- The customer's business is *disrupted*.
- The missing features were *clearly specified* in the Contract Software Requirements Specification.
- The customer *successfully sues* the developer for damages.

Physical Configuration Audit

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

Why PCA?



- A steel company installs a *complex control system* in their rolling mill.
- *Seven years later* the computer hardware platform ceases to be supported by its supplier.
- The company initiates a project to *refactor the software* for a new hardware platform.
- The source code on file does *not match* the executables running on the operational system. Further, the requirements specifications and design descriptions have *been lost*.

Performing Configuration Audits

- Assess the integrity of the **baselines**.
- Confirm that the configuration management **records** correctly identify the configuration items.
- Review the structure and integrity of the **items** in the configuration management system.
- 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.
- Confirm **compliance** with applicable configuration management standards and procedures.
- Track **action items** from the audit to closure.

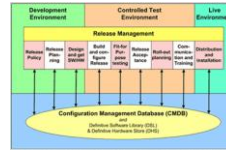


Outputs:

- Configuration audit results
- Action items

Release Management

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

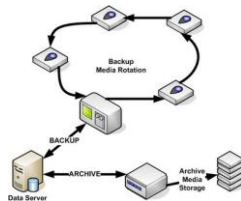


- Identify baselines
- Create Configuration Status Accounting Reports (CSAR)
- Perform PCA, FCA
- Package files and documents
- Receive client's confirmation

Backup And Archiving

Benefits:

- Recovery data when there are problems.
- Support version control.



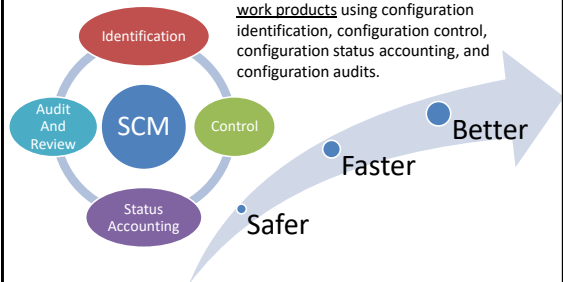
When project is finished:

- Archive data of project
- Archive or remove paper documents
- Clean all information of project

- Data archiving** is the process of moving data that is no longer actively used to a separate data storage device for long-term retention.
- Data archives consist of older data that is still important and necessary for **future reference**, as well as data that must be retained for **regulatory compliance**.

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

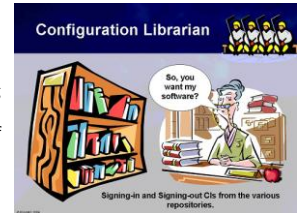
The person responsible for configuration management *implements*, maintains, and improves configuration management within the framework provided by management.



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

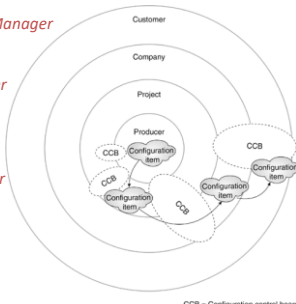
Librarian

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

CCB = Configuration control board

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

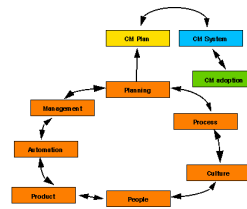
Deliver event registrations or change requests (depending on the form of cooperation) to the subcontractor



Configuration Management Plan



Configuration Management Plan



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.

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

✓ 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 (I)



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 (II)



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 (III)



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 (I)

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 (II)

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 (III)

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 (IV)

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

- Change reporting documentation

- 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:

- Charter
- Members
- Role
- Procedures
- 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 (V)

3.3 Configuration Status Accounting

3.3.1 Storage, handling and release of project media

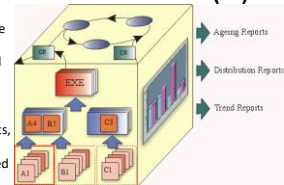
3.3.2 Types of information needed to be reported and the control over this information that is needed

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:

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

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



Outline of a Model CM Plan (VI)

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:

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

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

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



Outline of a Model CM Plan (VII)

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



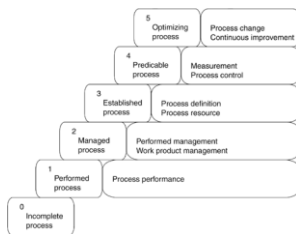


Configuration Management Solution



ISO 15504 (SPICE) [4]

Software Process Improvement and Capability Determination



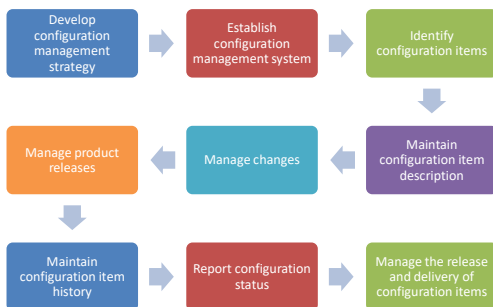
Work product management means that for any given process area to obtain level 2, all **relevant work products** from the performance of the process area must be placed under configuration management.

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



Standard Definition of CM

Standard definition (IEEE 729-1983)



Identification

- identifying components, structure
- What version of the file is this?

Control

- controlling releases and changes
- What changes went into the latest version of this product?

Status accounting

- recording, reporting status
- How many files were affected by fixing this one bug?

Audit and review

- validating completeness
- Are all the correct versions of files used in this current release?

Broaden Definition of CM

Based on existing CM systems, broaden definition

Manufacture

- managing construction, building
- What versions of files and tools were used to generate this latest release?

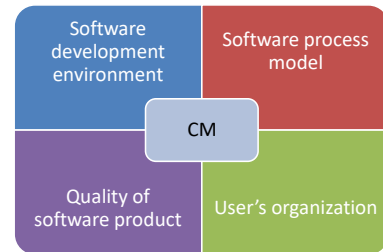
Process modelling

- ensuring life-cycle model
- Were all the files tested and checked for quality before being released to the customer?

Team work

- controlling team interactions
- Were all the locally made changes of the programmers merged into the last release of the product?

Effects



CM Solution

Planning

- Capturing all the important aspects about doing CM

Defining a process

- Capturing all the steps, tasks, and associated policies and procedures needed for doing CM

Dealing with people

- Catering for different user roles that exist in the organization

Automating support

- Helping with maintaining integrity and quality of the process and product

Making management decisions

- Deciding when to start using CM, whether to buy or build a CM system and how to best perform technology transition for the CM system

Perspectives

Corporate

- organization's view and process of CM

Project

- each project group may use a different CM system

Developer

- entails the specific functionality provided by a CM system

Application

- how CM is applied to a specific problem

In-house

Manual procedures and policies

Compiling code

Tracking and dealing with bugs

Version control

Third-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 & See You Again

