Configuration Management

Configuration Management (CM) is the process of controlling and documenting change to a system.

‘CM is a foundation of a project. Without it, no matter how talented the staff, how large the budget, how technically superior the development tools, project discipline will collapse, and success will be left to chance.’

Source: SPMN, Little Book of Configuration Management.

www.spmn.com

CM as a Speed Bump
Content

1. Objective of configuration management (CM)
2. Why bother with Software Configuration Management?
3. Functions of CM
4. Configuration Control Board
5. Role of QA
6. Bad excuses related to CM

Adapted from:

Sources of Changes

- Requirements.
  – Note: The longer the delivery cycle, the more likely requirements will change (e.g. 1% per month).
- Changes in funding.
- Technology advancements.
- Solutions to problems.
- Scheduling constraints.
- Customer expectations.
- Unexpected opportunities for an improved system.

Why Bother With Software CM?

1. **Software is easy to change**
   - Uncontrolled change is a common source of project chaos, schedule slippages and quality problems.

2. **Software is invisible**

3. **Types of components to keep trace**
   - Non executable components
     - Documentation
   - Executable components
     - Code
   - Tools used to develop and test software products

4. **Software is often changed during the life cycle**
   - **IKIWISE** (I’ll Know It When I See It)
     - Customer is not sure what they want
   - Scope creep (i.e. additional requirements)
   - Rework (i.e. because of errors)

---

Why Bother With SCM?

- **Examples of poor practices:**
  1. The latest version of source code cannot be found.
  2. A difficult defect that was fixed at great expense suddenly reappears.
  3. A developed and tested feature is mysteriously missing.
  4. A fully tested program suddenly does not work.
  5. The wrong version of the code was tested.
  6. There is no traceability between the software requirements, documentation, and code.
  7. Programmers are working on the wrong version of the code.
  8. The wrong versions of the configuration items are being baselined, delivered or installed in a product
  9. No one knows which modules comprise the software system delivered to the customer.

Benefits of CM to a Project

1. Reduces confusion and establishes order.
2. Organizes the activities necessary to maintain product integrity.
3. Ensures correct product configurations.
4. Limits legal liability by providing a record of actions.
5. Reduces life-cycle costs.
6. Enables consistent conformance with requirements.
7. Provides a stable working environment.
8. Enhances compliance with standards.

**Distribution of Rework During Development**

- **Rework 44%**
- **Production 56%**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Rework</th>
<th>Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Prelim.</td>
<td>6%</td>
<td>1%</td>
</tr>
<tr>
<td>Design</td>
<td>12%</td>
<td>4%</td>
</tr>
<tr>
<td>Detailed Design</td>
<td>16%</td>
<td>8%</td>
</tr>
<tr>
<td>Code &amp; Unit Test</td>
<td>12%</td>
<td>8%</td>
</tr>
<tr>
<td>Integr. &amp; System Test</td>
<td>19%</td>
<td>10%</td>
</tr>
</tbody>
</table>

Data from IEEE

---

**Terminology: Configuration Item (CI)**

- CI is the most *elementary entity* that will be *identified* and *managed* during the lifecycle.
  - Élément de configuration

- **A set of CIs**
  - Baseline (référentiel)
    - An approved *snapshot* of the system at *appropriate points* in the development lifecycle
      - Serves as the *basis* for further development
        - e.g. Entry criteria for next phase
      - Can be changed only through an agreed upon *change procedure*
  - SCI – Software Configuration Item
Terminology: Baselines

- For Software Engineering (CMMI)
  - **Baseline**.
    - A set of requirements, design, source code files and the associated executable code, build files, and user documentation that have been assigned a unique identifier.
  - Release of a baseline constitutes retrieval of source code files (configuration items) from the configuration management system and generating the executable files.
  - A baseline that is delivered to a customer is typically called a “release”
  - A baseline for an internal use is typically called a “build.”

System Life Cycle Baselines

Adapted from DoD Standard 2167A
Mapping of system and developmental baselines

- Functional baseline → System requirements specification review
- Allocated baseline → Software requirements specification review
- Requirement baseline → Software requirements specification review
- General design baseline → Interface requirements specification review
- Detailed design baseline → Design specification review
- Module baselines → Code walkthroughs or inspections
- Integration baselines → Module testing from baseline
- Operational baseline → Integration and system testing from baseline
- Physical configuration audit on operational baseline and deliverable customer documentation

Kasse, T., McQuaid, P., ‘Software Configuration Management for Project Leaders’
Purpose of Configuration Management

- To establish and maintain the integrity of work products using:
  1. configuration identification,
  2. configuration control,
  3. configuration status accounting,
  4. configuration audits.

Four Key Functions of CM

Functions of CM
1. Configuration Identification

- **Role:** To define baseline components
- **Question:** *What* is my system configuration?
  - Your system configuration consists of the following items:
    Item1, Item2, Item3,…
- **Major Activities**

```
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify Requirements</td>
<td>Required to establish the baseline components</td>
</tr>
<tr>
<td>Identify Configuration Items</td>
<td>Confirm the system configuration</td>
</tr>
<tr>
<td>Define Baselines</td>
<td>Establish the identification scheme</td>
</tr>
<tr>
<td>Document Configuration Items</td>
<td>Requirements, Identification Scheme, Baselines</td>
</tr>
</tbody>
</table>
```

Critères pour sélectionner les CI ? Exercice 1, 2

Items Under CM Control

<table>
<thead>
<tr>
<th>Items Under CM Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>System data files</td>
</tr>
<tr>
<td>Requirements specifications</td>
</tr>
<tr>
<td>Design specifications</td>
</tr>
<tr>
<td>Test plans</td>
</tr>
<tr>
<td>Test data sets</td>
</tr>
<tr>
<td>User documentation</td>
</tr>
<tr>
<td>Quality plans</td>
</tr>
<tr>
<td>Compilers</td>
</tr>
<tr>
<td>Debuggers</td>
</tr>
<tr>
<td>Shell scripts</td>
</tr>
<tr>
<td>Other related support tools</td>
</tr>
<tr>
<td>Development procedures and standards [4]</td>
</tr>
</tbody>
</table>

Functions of CM
2. Configuration Control

- **Role:** To provide a mechanism (i.e. documentation, organizational body, procedures) for preparing, evaluating, approving or disapproving all changes throughout the lifecycle (CCB)
  - Example of a document: Engineering Change Proposal (ECP)

- **Question:** How do I control changes to my configuration?
  - The steps in processing changes are: Step1, Step2,…

---

Configuration Control
Basic Ingredients

1. **Process and Procedures** for controlling changes
2. **An organizational body**
   - For formally evaluating and approving or disapproving a proposed change
     - E.g. Configuration Control Board (CCB)
3. **Documentation** for formally initiating and defining a proposed change
   - Deficiencies, hardware change, new requirement, economic savings, schedule accommodation
   - E.g. Engineering Change Proposal (ECP)
Incidents

- An incident is every (significant) unplanned event observed, and requiring further investigation.
  - A wrong formulation in a document,
  - A coding mistake found during a walkthrough,
  - An enhancement request arising from a new idea from the customer,
  - A change required in the code because of the upgrade to a new version of the middleware supporting the system.

- All incidents must be registered.

Change Control Phases

<table>
<thead>
<tr>
<th>Phase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creation of the incident registration.</td>
<td>The incident is described in the registration.</td>
</tr>
<tr>
<td>Analysis of the incident registration.</td>
<td>It is determined which configuration item(s) will be affected by possible changes, and the change effort is estimated.</td>
</tr>
<tr>
<td>Rejection or acceptance of the incident.</td>
<td>If the incident is accepted, a change request is created for each of the affected configuration items.</td>
</tr>
<tr>
<td>The change request initiates a new configuration item.</td>
<td>A new configuration item is identified and created, and the change is implemented. In the course of acceptance and placement in storage of the new configuration item, feedback is given to the CCB.</td>
</tr>
<tr>
<td>Closing of the change request.</td>
<td>The change request can be closed when the change has been implemented and accepted.</td>
</tr>
<tr>
<td>Closing of the incident registration.</td>
<td>The incident registration can be closed when all the corresponding change requests are closed.</td>
</tr>
</tbody>
</table>

Elements of Change Control Process

1. Who can initiate the change requests
2. The individuals, group, or groups who are responsible for evaluating, accepting, and tracking the change proposals for the various baselined products
3. What the criteria are for placing the software components under formal change control
4. The "change impact" analysis expected for each requested change
5. How revision history should be kept
6. The check-in/check-out procedures
7. The process the software configuration control board (SCCB) follows to approve changes
8. How change requests will be linked to the trouble-reporting system
9. How change requests are tracked and resolved
10. The reviews and/or regression tests that must be performed to ensure that changes have not caused unintended effects on the baseline
11. The procedure that will be followed to update all affected software.

Flow of a Change Request

Impact Analysis Report Template.

Change Request ID: ______________ Title: ______________
Description: ______________ Analyst: ______________
Date Prepared: ______________
Prioritization Estimates:
  Relative Benefit: (1-9) __
  Relative Penalty: (1-9) __
  Relative Cost: (1-9) __
  Relative Risk: (1-9) __
Calculated Priority: ______________
Estimated total effort: ______________ labor hours
Estimated lost effort: ______________ labor hours
Estimated schedule impact: ______________ days
Additional cost impact: ______________ dollars
Quality impact: ______________
Other requirements affected: ______________
Other tasks affected: ______________
Plans to be updated: ______________
Integration issues: ______________
Life cycle cost issues: ______________

Configuration Control Board (CCB)

- A body that provides the means to implement change control at optimum levels of authority.
  - Customer’s CCB
  - Prime contractor’s/integrator’s CCB
    - Developer’s CCB
  - Subcontractors’ CCB
- A body of people (an individual or a group) that is empowered to
  - Make binding decisions about which proposed changes and suggested features to incorporate into the product.
CCB - Membership

1. Project management
2. Development
3. Testing
4. Quality Assurance
5. Customer representatives
6. User documentation developers
7. Technical support
8. Configuration management

CCB - Decision Making Process

- **Identify:**
  1. The number of CCB members or the key roles that constitutes a quorum for making decisions.
  2. Whether voting, consensus, unanimity, or some other mechanism is used to make decisions.
  3. Whether the CCB Chair is permitted to overrule the CCB’s collective decision.
  4. Whether a higher level CCB or someone else must ratify the decision.
CCB - Decision Making Criteria

1. Financial savings or revenue
2. Customer satisfaction,
3. Competitive advantage,
4. Time to market.
5. Product cost
6. Delivery schedule,
7. Product quality
8. Product functionality,
9. Support costs,
10. User dissatisfaction.

11. Other criteria...

Change Request Flow with CCB

- **Draw a flowchart showing:**
  1. Each stage of a Change Request (CR)
  2. The status of the CR at each stage
     - E.g. submitted, approved

**Note:** The CCB is performing its functions

Exercise
Change Request Flow with CCB

Submitted

Originator submits a change request

Evaluator performs impact analysis

Evaluated

Evaluator decides to make the change

Verified

Verification (e.g. regression test) failed

Verified

Modified product is installed

Closed

Change request is archived

CCB

Wiegers K., Software Requirements, Microsoft Press, 2003

Change Request Flow – Swimlane Notation

PM

Engineering

CCB

Procurement

Manufacturing

6/21/2008
Change Request Flow for Safety Critical Software

Categorization of Changes

- The basis for assigning a change category is its potential impact on the Global Risk Profile of Rail-Transit operations.
- Changes are categorized as follows:
  - **Category A**: A change which will definitely impact a safety critical system, sub-system, or operating practice (e.g. operator training) and requires subsequent review by the Safety Review Committee if approved by the CCB.
  - **Category B**: A change which may impact a safety critical system, sub-system or operating practice, either directly or through an interface and therefore requires evaluation and formal recategorization by CCB as a Category A or Category C change.
  - **Category C**: Changes which have no impact on the system safety profile of operations and which can therefore be recommended for implementation by the CCB.

Types of Changes

- The deployment and operation of the Rail-Transit system in some cases require that contractual considerations be included in CM process.
- Changes are classified by type as follows:
  - **Type I**: Has no contractual implications
  - **Type II**: Has contractual implications which are technical in nature only; has no commercial implications.
  - **Type III**: Has contractual implications which are commercial in nature only; has no technical implications.
  - **Type IV**: Has both technical and commercial implications with respect to the contract.
Change Request Flow for Safety Critical Software

Source: A.E. Fazio N.S. Shashidhara, Configuration Management for Rail-Transit Operations, 21st Century Rail Corporation Jersey City, NJ.

Functions of CM
3. Configuration Status Accounting

Question:
- **What** changes have I made to the system?
  - Your configuration and related changes at this time are: (Item1, Item2) + (Change1, Change2) + (Pending Change1, …)
  - **What** changes remain to be implemented?
Functions of CM
3. Configuration Status Accounting

- **Role**
  1. **Provide a mechanism for maintaining a record of the evolution of a system (e.g. history) at any time:**
     1. Is this specification **approved**?
     2. Is this subsystem **tested**?
     3. Has a **change request** been approved or rejected by the SCCB?
     4. Which **version** of an item implements an approved change request?
     5. What is **different** about a new version of a system?
     6. How many **faults** are detected each month and how many are **fixed**?
     7. What is the **cause** of the trouble report?
  2. **Report on the traceability of all changes to the baseline throughout the software lifecycle**

**Exercice**

---

Functions of CM
4. Configuration Auditing

**Question**: Does the system I am building satisfy the stated needs?

Your system, as currently built, differs from the stated needs as follow: Difference1, Difference2, …
Functions of CM
4. Configuration Auditing

- **Roles**
  1. **Provide a mechanism** for determining the degree to which the current state of the system mirrors the system pictured in baseline and requirements documentation
     - Verify that the software product is built according to the requirements, standards, or contractual agreement
     - Verify that all software products have been produced, correctly identified and described, and that all change requests have been resolved
  2. **Provide the mechanism for establishing a baseline**
  3. **Ensure SCM process and procedures are performed**

---

Functional Configuration Audits (FCA)

- **Objective**
  - To provide an independent evaluation of the software products, verifying that each CI’s actual functionality and performance is consistent with the software requirements specification.

- **An FCA includes:**
  1. An audit of the formal test documentation against test data
  2. An audit of the verification and validation reports to ensure their accuracy
  3. A review of all approved changes (problem reporting and corrective actions) to ensure they have been correctly technically incorporated and verified
  4. A review of the updates to previously delivered documents to ensure their accuracy and consistency
  5. A sampling of the design review outputs to ensure that all findings were completed and incorporated
  6. A comparison of the code with the documented software requirements to ensure that the code addresses all and only the documented requirements
  7. A review to ensure that all testing was accomplished with appropriate test documentation and approved test data to ensure that all configuration items meet the established performance criteria
Physical Configuration Audit (PCA)

- **Objective**
  - To provide an independent evaluation of the system CIs to confirm that each item that makes up the “as built” system maps to its specifications.
  - To verify that the software and its documentation are internally consistent and ready for delivery to the customer or end user.
- **A PCA includes:**
  1. An audit of the system specification for completeness
  2. An audit of the FCA report for discrepancies and actions taken
  3. A comparison of the architectural design with the detailed design components for consistency
  4. A review of the module listing for compliance with approved coding standards
  5. An audit of the manuals for format completeness and conformance to systems and functional descriptions. Such manuals include user’s manuals, programmer’s manuals, and operator’s manuals

Traceability Audits

- To verify throughout the software development process:
  1. Can the software requirements be traced back to the system requirements allocated to software?
  2. Can the architectural design be traced back to the software requirements?
  3. Can the detailed design(s) be traced back to the architectural requirements?
  4. Can the code modules be traced back to detailed design modules?
  5. Can the module tests, integration tests, and system tests be traced back to the software requirements?
FCA and PCA


Software Library System

1. Supports multiple control levels of SCM
   - Managers, developers, SCM, QA, Systems engineering, etc.
2. Provides for the storage and retrieval of configuration items
3. Provides for the sharing and transfer of configuration items among affected groups and among control levels within the library
4. Provides for the storage and recovery of archive versions of configuration items
5. Provides service functions
   - checking status, verifying the presence of all built items, and integrating changes into a new baseline
6. Ensures the correct creation of products from the software baseline library
7. Provides for the storage, update, and retrieval of SCM records
8. Supports the production of SCM reports
9. Supports the tracing of requirements, forward and backward, throughout the life cycle
10. Provides safe and secure storage for configuration items (key project components) so they cannot be changed without authorization.
Processes needed for CM by CMMI

<table>
<thead>
<tr>
<th>For Establishing Baselines:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Convention(s) for unique identification.</td>
</tr>
<tr>
<td>• Convention(s) for identification of single items and composite items.</td>
</tr>
<tr>
<td>• Procedure(s) for registration of information about each configuration item.</td>
</tr>
<tr>
<td>• Procedure(s) for placement in storage and related update of information.</td>
</tr>
<tr>
<td>• Procedure(s) for release of items.</td>
</tr>
<tr>
<td>• Template(s) for item approval registration.</td>
</tr>
<tr>
<td>• Template(s) for release request.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>For Tracking and Controlling Changes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Convention(s) for formation of different types of CCBs.</td>
</tr>
<tr>
<td>• Definition of responsibility for each type of CCB.</td>
</tr>
<tr>
<td>• Description of the change control process.</td>
</tr>
<tr>
<td>• Procedures forming the life cycles for incidents and changes.</td>
</tr>
<tr>
<td>• Template(s) for incident registration.</td>
</tr>
<tr>
<td>• Template(s) for change request.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>For Establishing Integrity:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Procedure(s) for production of available reports.</td>
</tr>
<tr>
<td>• Procedure(s) for ad-hoc extracts of information.</td>
</tr>
<tr>
<td>• Procedure for audit.</td>
</tr>
<tr>
<td>• Templates for the reports that the CM system is expected to produce.</td>
</tr>
</tbody>
</table>

Subcontractor Control

• Ensures that the subcontractor is able to maintain the integrity of the subsystem it has contracted for, including:
  1. Placing necessary life-cycle products under configuration control to ensure consistency with the main development effort
  2. Maintaining a subcontractor software library that will release the agreed upon configuration items or subsystems to the contracting organization
Sample Chart of Requirements Change Activity

Number of Requirements Changes

Weeks After SRS was Baselined

0 5 10 15 20

0 2 4 6 8 10 12 14 16

Sample chart of requirements change Origins

Number of Requirements Changes

Change Origin

Marketing
Management
Customer
Software Eng.
Hardware Eng.
Testing
Evolution of Requirements Status

- Coverage ratio of higher level requirements
- Coverage ratio of requirements by associated requirement tests

Change Control Policy -1

- **Elements of a change control policy:**
  1. All requirement changes must follow the process.
     - If a change request is not documented according to this procedure, it will not be acted upon.
  2. The change control board (CCB) designated for each project will decide which changes to implement.
     - Simply requesting a change does not guarantee that it will be made.
  3. The contents of the change database shall be visible to all project stakeholders.
Change Control Policy -2

• Elements of a change control policy:

4. The original text of a change request shall not be modified or deleted from the database.

5. Every incorporated requirement change shall be traceable back to an approved change request.

6. No design or implementation work will be performed on unapproved requirement changes other than feasibility exploration to assist with making the approval decision.

Typical SCM Plan

1. The scope of the plan, including the project, the software to be developed, and the life-cycle phases
2. The relationship between the SCM plan and the other standards or plans that describe how the project will be managed
   • e.g. software development, SQA plan
3. SCM roles and responsibilities
4. Configuration identification
5. Baselining
6. Configuration control
7. Configuration management status accounting
8. Interface control
9. Subcontractor control
10. Software configuration audits
11. Software library

IEEE Standard 828 – 2005
Configuration Management Plans
Real-World Considerations

1. Management Commitment
   – Essential to achieving benefits of CM

2. CM Staffing
   – Enough experienced people

3. Tooling
   • See ISO CM Tool Requirements Std

4. Establishment of a CCB
   – A starting point in instituting change control

5. Avoiding the Paperwork Nightmare
   – Stakeholders should agree on the amount of paperwork needed to achieve mutual level of visibility and traceability

Role of QA

1. Audit the software development activities and product
   • i.e., Functional Configuration Audits (FCA) and Physical Configuration Audits (PCA)

2. Certify CM compliance with Project plan, CM Plan, standard, processes and procedures.

3. Participate in the CCB and Interface Control Working Group
**List of Some Bad Excuses Related to CM**

1. CM only applies to **documents**
2. CM only applies to **code**
3. CM applies only to **military** and **aerospace** projects
4. Our **tool** does all the CM for us
5. If we have CCB, we won’t be able to **do any work around here**
6. I don’t do CM, it applies **only to big projects**
7. You can’t **stop technical people** from making a quick change when they find a problem
8. CM applies only to our **subcontractors**
9. CM will raise our **costs**

*Source: SPMN, Little Book of Bad Excuses.*

---

**Basic Configuration Management**

- **Simple steps** will add control and project tracking information:
  1. Uniquely **identify** system components
  2. Formalize the use of **reviews** before a configuration item is **baselined**
  3. Establish simple **change control**
  4. Build up a **repository**
     - configuration items
     - change requests
     - problem reports
  5. **Restrict/Control access** to the project library
Project Management Institute

- Practice Standard for Project Configuration Management
- www.pmi.org

Summary

1. Objective of configuration management (CM)
2. Why bother with Configuration Management?
3. Functions of CM
4. Configuration Control Board
5. Bad excuses related to CM
Questions