

Configuration Management

Special:UserLogin > Configuration Management

The printable version is no longer supported and may have rendering errors. Please update your browser bookmarks and please use the default browser print function instead.

Lead Authors: Ray Madachy, John Snoderly, Garry Roedler, **Contributing Author:** Rick Adcock

The purpose of configuration management (CM) is to establish and maintain the integrity of all of the identified outputs of a project or process and make them available to concerned parties (ISO/IEC/IEEE 2015). Unmanaged changes to system artifacts (such as those associated with plans, requirements, design, software, hardware, testing, and documentation) can lead to problems that persist throughout the system life cycle. Hence, one primary objective of CM is to manage and control the change to such artifacts.



Contents

Configuration Management Process Overview

- Planning

- Configuration Identification

- Establishing Baseline

- Change Control

- Configuration Auditing

- Constraints and Guidance

- Organizational Issues

- Measurement

- Tools

Linkages to Other Systems Engineering Management Topics

Practical Considerations

Pitfalls

Good Practices

References

Works Cited

Primary References

Additional References

Configuration Management Process Overview

CM is the discipline of identifying and formalizing the functional and physical characteristics of a configuration item at discrete points in the product evolution for the purpose of maintaining the integrity of the product system and controlling changes to the baseline. The baseline for a project contains all of the technical requirements and related cost and schedule requirements that are sufficiently mature to be accepted and placed under change control by the project manager. The project baseline consists of two parts: the technical baseline and the business baseline. The systems engineer is responsible for managing the technical baseline and ensuring that it is consistent with the costs and schedules in the business baseline. Typically, the project control office manages the business baseline.

The ANSI/GEIA EIA-649-A standard presents CM from the viewpoint that configuration management practices are employed because they make good business sense rather than because requirements are imposed by an external customer (ANSI/GEIA 2005). The standard discusses CM principles and practices from an enterprise view; it does not prescribe which CM activities individual organizations or teams within the enterprise should perform. Each enterprise assigns responsibilities in accordance with its own management policy. See also the Implementation Guide for Configuration Management, which supports and provides further information on this standard (ANSI/GEIA October 2005).

Effective CM depends on the establishment, maintenance, and implementation of an effective process. The CM process should include, but is not limited to, the following activities:

- identification and involvement of relevant

stakeholders

- setting of CM goals and expected outcomes
- identification and description of CM tasks
- assignment of responsibility and authority for performing the CM process tasks
- establishment of procedures for monitoring and control of the CM process
- measurement and assessment of the CM process effectiveness

As a minimum the CM process should incorporate and detail the following tasks (SEI 2010):

- identifying the configuration of selected work products that compose the baselines at given points in time
- controlling changes to configuration items
- building or providing specifications to build work products from the configuration management system
- maintaining the integrity of baselines
- providing accurate status and current configuration data to developers, end users, and customers

Figure 1 below shows the primary functions of systems CM.



Figure 1. Configuration Management Functions. (SEBoK Original)

Planning

The CM plan must be developed in consideration of the organizational context and culture; it must adhere to or incorporate applicable policies, procedures, and standards and it must accommodate acquisition and subcontractor situations. A CM plan details and schedules the tasks to be performed as part of the CM process including: configuration identification, change control, configuration status accounting, configuration auditing, and release management and delivery.

Configuration Identification

This activity is focused on identifying the configuration items which will be managed and controlled under a CM process. The identification activity involves establishing a procedure for labeling items and their versions. The labeling provides a context for each item within the system configuration and shows the relationship between system items.

Establishing Baseline

Configuration items are typically assembled into a baseline which specifies how a system will be viewed for the purposes of management, control, and evaluation. This baseline is fixed at a specific point in time in the system life cycle and represents the current approved configuration. It generally can only be changed through formal change procedures.

Change Control

A disciplined change control process is critical for systems engineering. A generalized change control process in response to an engineering change proposal (ECP) is shown in Figure 2 below, which is adapted from Systems Engineering and Analysis (Blanchard and Fabrycky 1999).

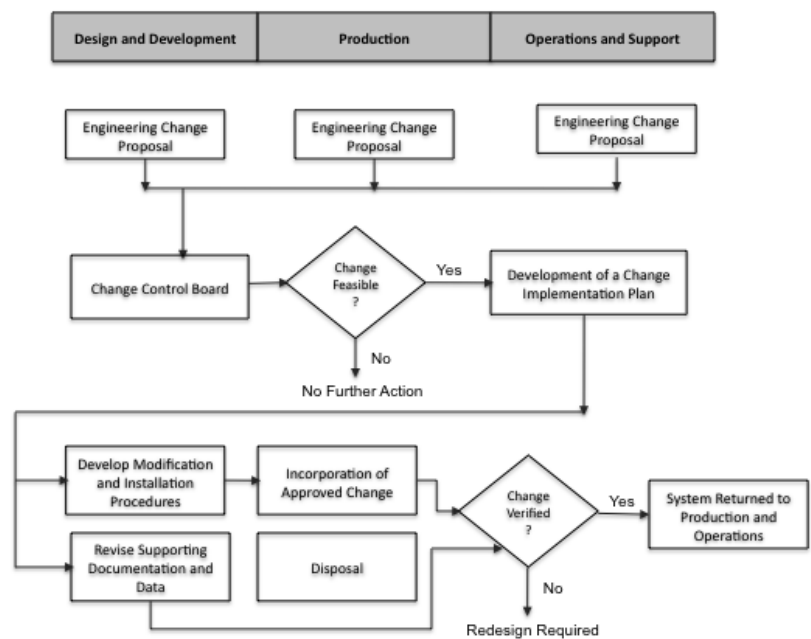


Figure 2. Configuration Change Control Process. (SEBoK Original)

Configuration Auditing

Audits are independent evaluations of the current status of configuration items and determine conformance of the configuration activities to the CM process. Adherence to applicable CM plans, regulations, and standards, is typically assessed during audits.

Constraints and Guidance

Constraints affecting and guiding the CM process come from a number of sources. Policies, procedures, and standards set forth at corporate or other organizational levels might influence or constrain the design and implementation of the CM process. Also, the contract with an acquirer or supplier may contain provisions affecting the CM process. The system life cycle process adopted and the tools, methods, and other processes used in system development can affect the CM process (Bourque and Fairley 2014). There are a variety of sources for guidance on the development of a CM process. These include the ISO standards on system life cycle processes (ISO/IEC/IEEE 15288 2015) and configuration management guidelines (ISO 10007 2003), as well as the *Guide to The Software Engineering Body of Knowledge* (SWEBOK) (Bourque and Fairley 2014), and the CMMI for Development (SEI 2010).

Organizational Issues

Successful CM planning, management, and implementation requires an understanding of the organizational context for the design and implementation of the CM process and why constraints are placed upon it. To plan a CM process for a project, it is necessary to understand the organizational context and the relationships among the organizational elements. CM interacts with other organizational elements, which may be structured in a number of ways. Although the responsibility for performing certain CM tasks might be assigned to other parts of the organization, the overall responsibility for CM often rests with a distinct organizational element or designated individual (Bourque and Fairley 2014).

Measurement

In order to carry out certain CM functions, such as status accounting and auditing, as well as to monitor and assess the effectiveness of CM processes, it is necessary to measure and collect data related to CM activities and system artifacts. CM libraries and automated report tools provide convenient access and facilitation of data collection. Examples of metrics include the size of documentation artifacts, number of change requests, mean time to change to a configuration item, and rework costs.

Tools

CM employs a variety of tools to support the process, for example:

- library management
- tracking and change management
- version management
- release management

The INCOSE Tools Database Working Group (INCOSE TDWG 2010) maintains an extensive list of tools including configuration management.

Linkages to Other Systems

Engineering Management Topics

Configuration management is involved in the management and control of artifacts produced and modified throughout the system life cycle in all areas of system definition, system realization, system deployment and use, and product and service life management. This includes CM application to the artifacts of all the other management processes (plans, analyses, reports, statuses, etc.).

Practical Considerations

Key pitfalls and good practices related to systems engineering CM are described in the next two sections.

Pitfalls

Some of the key pitfalls encountered in planning and performing CM are in Table 1.

Table 1. Configuration Management Pitfalls. (SEBoK Original)

Name	Description
Shallow Visibility	• Not involving all affected disciplines in the change control process.
Poor Tailoring	• Inadequate CM tailoring to adapt to the project scale, number of subsystems, etc.
Limited CM Perspective	• Not considering and integrating the CM processes of all contributing organizations including COTS vendors and subcontractors.

Good Practices

Some good practices gathered from the references are provided in Table 2 below.

Table 2. Configuration Management Good Practices.
(SEBoK Original)

Name	Description
Cross-Functional CM	• Implement cross-functional communication and CM processes for software, hardware, firmware, data, or other types of items as appropriate.
Full Lifecycle Perspective	• Plan for integrated CM through the life cycle. Do not assume that it will just happen as part of the program.
CM Planning	• Processes are documented in a single, comprehensive CM plan early in the project. The plan should be a (systems) CM plan. • Include tools selected and used.
Requirements Traceability	• Initiate requirements traceability at the start of the CM activity.
CCB Hierarchy	• Use a hierarchy of configuration control boards commensurate with the program elements.
Consistent Identification	• Software CI and hardware CI use consistent identification schemes.
CM Automation	• Configuration status accounting should be as automated as possible.

Additional good practices can be found in ISO/IEC/IEEE (2009, Clause 6.4) and INCOSE (2010, sec. 5.4.1.5).

References

Works Cited

ANSI/GEIA. 2005. Implementation Guide for Configuration Management. Arlington, VA, USA: American National Standards Institute/Government Electronics & Information Technology Association, GEIA-HB-649. October 2005.

Blanchard, B.S. and W J. Fabrycky. 2005. *Systems Engineering and Analysis*, 4th ed. Prentice-hall international series in industrial and systems engineering. Englewood Cliffs, NJ, USA: Prentice-Hall.

Bourque, P. and R.E. Fairley (eds.). 2014. *Guide to the Software Engineering Body of Knowledge (SWEBOK)*.

Los Alamitos, CA, USA: IEEE Computer Society.
Available at: <http://www.swebok.org>

ISO. 2003. *Quality Management Systems - Guidelines for Configuration Management*. Geneva, Switzerland: International Organization for Standardization (ISO), ISO 10007:2003.

ISO/IEC/IEEE. 2015. *Systems and Software Engineering-- System Life Cycle Processes*. Geneva, Switzerland: International Organisation for Standardisation / International Electrotechnical Commissions. ISO/IEC/IEEE 15288:2015

SEI. 2010. *Capability Maturity Model Integrated (CMMI) for Development*, version 1.3. Pittsburgh, PA, USA: Software Engineering Institute (SEI)/Carnegie Mellon University (CMU).

Primary References

ANSI/EEIA. 2016. *Configuration Management Standard Implementation Guide*. Arlington, VA, USA: American National Standards Institute/Government Electronics & Information Technology Association, GEIAHB649A. February 2016.

ANSI/GEIA. 2005. *Implementation Guide for Configuration Management*. Arlington, VA, USA: American National Standards Institute/Government Electronics & Information Technology Association, GEIA-HB-649. October 2005.

GEIA. 2004. *GEIA Consensus Standard for Data Management*. Arlington, VA, USA: Government Electronics & Information Technology Association, GEIA-859.

ISO. 2003. *Quality Management Systems - Guidelines for Configuration Management*. Geneva, Switzerland: International Organization for Standardization (ISO), ISO 10007:2003.

ISO/IEC/IEEE. 2015. *Systems and Software Engineering-- System Life Cycle Processes*. Geneva, Switzerland: International Organisation for Standardisation / International Electrotechnical Commissions. ISO/IEC/IEEE 15288:2015

SEI. 2010. *Capability Maturity Model Integrated (CMMI) for Development*, version 1.3. Pittsburgh, PA, USA: Software Engineering Institute (SEI)/Carnegie Mellon

University (CMU).

Additional References

INCOSE Tools database working group (TDWG). in International Council on Systems Engineering (INCOSE) [database online]. San Diego, CA, USA, 2010. Accessed April 13, 2015. Available at: <http://www.incose.org/docs/default-source/wgcharters/tools-database.pdf>.

INCOSE. 2008. "INCOSE measurement tools survey." in International Council on Systems Engineering (INCOSE) [database online]. San Diego, CA, USA, 2008.

ISO/IEC/IEEE. 2009. *Systems and Software Engineering - Life Cycle Processes - Project Management*. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronics Engineers (IEEE), ISO/IEC/IEEE 16326:2009(E).

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.10, released 06 May 2024

Retrieved from

"https://sandbox.sebokwiki.org/index.php?title=Configuration_Management&oldid=71142"

This page was last edited on 2 May 2024, at 21:54.