# CONFIGURATION MANAGEMENT (CM) GUIDELINES

Prepared by: S. E. Snook
Software Quality Engineering
330 Corporate Way
Suite 300
Orange Park, FL, USA 32073

# 1. PURPOSE STATEMENT

This guideline describes recommended guidelines to follow in implementing configuration management for project configuration items. These items may include but are not limited to the following:

a. Developed project computer software configuration items (releases)
b. Project documentation and specifications
b. Web site content
c. Operational Procedures
d. Training data
e. Computer system support resources (hardware and support software)

Configuration Management (CM) is a discipline which applies technical and administrative direction, surveillance, and control to all project configuration items. The scope of configuration management addressed in this guidelines has been developed using *IEEE 1042-1987 Guide to Software Configuration Management (ANSI)*, and *IEEE 828-1983 Standard for Software Configuration Management Plans* (ANSI) as guidelines.

Configuration Management includes the following:
a. Planning
b. Configuration item identification
c. Configuration control
d. Status accounting and reporting
f. Release to production process

The purpose of CM is to maintain the integrity of the product and engineering effort so that the contractual, functional and performance requirements of the system will be met as well as provide a disciplined baseline change control process.

One of the key functional components of configuration management is change/version control. Use of an automated tool is recommended to facilitate implementation of change control as well as other CM procedures. It is recommended that a specific written procedure be documented for use of any automated CM tool selected that incorporates the guidelines in this document.

# 2. GUIDELINES

### *INTRODUCTION*

Configuration management is the process of formally identifying and controlling project configuration items.
The following definitions apply to this set of guidelines:

## *2.1 INTRODUCTION*

Configuration management is the process of formally identifying and controlling project configuration items. The following definitions apply to this set of guidelines:

**Baseline** - A set of documents, specifications, and/or software products that have been formally reviewed and agreed upon, that thereafter serve as the basis for further development, and that can be changed only through formal change control procedures. A baseline is formally designated and fixed at specific times during the life cycle of a project configuration item or project. Baselines, plus approved changes from those baselines, constitute the current configuration identification.

**Computer Software Component (CSC)** - A functionally or logically distinct part of a computer software configuration item (CSCI), typically an aggregate of two or more computer software units (CSU).

**Computer Software Configuration Item (CSCI)** - The sum total or aggregation of the application software that is resident on a single computer or CPU. It is treated as a single entity in the configuration management process.

**Project/System** - The primary physical parts of a system are Hardware Configuration Item(s) (HWCI,) Computer Software Configuration Item(s) (CSCI) and documentation.

**Hardware Configuration Item (HWCI)** - The computer hardware required to run the Project/System application software.

**Release (Revision, Version) -** Any change to any project configuration item. Revision numbers have three levels, x, y, and z (e.g., revision 1.2.3 implies level x=l, level y=2, and level z=3). When x changes, y and z must be set to zero. (The terms *release*, *revision*, or *version* may be used interchangeably as long as they are consistent within a given project.)

## *2.2 CONFIGURATION ITEM IDENTIFICATION AND NUMBERING*

In order to have a consistent project CM process, nomenclature, and organization the following CSCI configuration item example is recommended for project configuration items:

As a general guideline, changing 10 percent or more of the respective baselined project configuration item constitutes a major change - new release (revision/version). Critical/emergency releases are a separate category of mandatory changes that are time-sensitive and cannot be postponed until the next planned release. All other changes are considered minor changes.

The following provides expanded details and examples:

a. Number the initial project configuration item Release/Revision/Version 1.0.0. Number major changes with sequential numbers in number level x (e.g., the first major change to Release/Revision/Version 1.0.0 is numbered 2.0.0).
b. Number minor changes to a project configuration item with sequential decimal numbers in number level y (e.g., the first minor change to Release/Revision/Version 1.0.0 is numbered 1.1.0, and the second minor change is numbered Version 1.2.0).
c. Number emergency changes to a project configuration item Release/Revision/Version with sequential decimal numbers in number level z (e.g., the first emergency change to Version 1.0.0 is numbered Version 1.0.1, and the second emergency change to Version 1.0.0 is numbered Version 1.0.2).

Computer Software Configuration Item (CSCI) Release example:

e.g., Release **1.0.0** => the first "**MAJOR**" release of a software project
Release **2.0.0** => the second "**MAJOR**" release of a software project; ~more than 10% of the previous release's functionality has been changed
Release 1.**1.0** => the first "**MINOR**" release of a previous "**MAJOR"** Release 1.0.0; ~less than 10% of the previous release's functionality has been changed
Release 2.**2.0** => the second "**MINOR**" release of a of a previous "**MAJOR**" Release 2.0.0; ~less than 10% of the previous release's functionality has been changed
Release 7.**x** =>some future undetermined "MINOR" release; most likely a "MINOR" release of a previous "MAJOR" release 7.n

Release 6.1.**1**=> The first "Emergency" Release of "MINOR" release 6.1.0; this implies a quick reaction development/testing/implementation for "show stopper" enhancements and "Incidents

Documentation project configuration item example:

e.g., Assign all documentation a unique title (for example, System Test Plan, Software Design Document), a release/revision/version date, and optionally a document release/revision/version number. The document title and date can be used as the simplest, complete and unique identifier of a document given that the data is changed each time ANY document content is changed. This uniquely identifies a release/revision/version of a document, however it gives no indication of the revision history. A release number like the 3 digit number used the above CSCI may be used.

### 2.2.2 CHANGE CONTROL

Change control may be implemented as follows:

a. All project configuration items are maintained in an electronic file or automated tool repository for which access is controlled.

b. Working copies of baselined project configuration items may be "checked out" to staff engineers to be used in preparing new draft releases/revisions/versions. Only the project leader or designee can give permission to check out these working copies of baselined project configuration items of their respective projects. These working copies are part of an approved baseline only after approval is received from the project leader or designee. If multiple copies of a project configuration item are "checked out" care must be taken to assure all changes are incorporated when checking these items back into the repository. It is not recommended that multiple copies of a given project configuration item be "checked out" concurrently.

c. Any changes to the currently approved baseline of a project configuration item are made after receiving any required reviews and/or approvals. At this point the respective project configuration item is "checked in" to the project repository.

### 2.2.3 STATUS ACCOUNTING AND REPORTING

Configuration status accounting is the process used to record and report the status of changes to project configuration items under formal configuration management.

For each project configuration item, the project leader should maintain an organized set of the configuration management records. The configuration management records should include the following:

a. Documentation records used to certify that project configuration items are ready for release for technical review or approval.

b. Documentation status records used to indicate project configuration items release, review, and approval schedule and status.

c. Status of project configuration items change proposals (e.g., Lotus Notes *Incidents*, *Enhancements*, *Issues*)

d. Communication/distribution of changes made and pending approval or implementation.

The project leader may submit a configuration management status report periodically to the designated department manager if any changes have been made to any respective project configuration item since the last report. This status report should contains the following:

a. A complete list of the latest project configuration items and associated release/revision/version numbers.

b. A summary statement of most recent modification(s) to each project configuration item; a brief description of changes from last report.

c. Dates any changes were incorporated

d. Life cycle development progress records indicating the status of each project configuration items (e.g., “being revised”, “final”, “draft”, etc.).