5000
Ser 06
FEB [illegible]

MEMORANDUM

From: Assistant Commander, Engineering
To: Distribution

Subj: CONFIGURATION MANAGEMENT-IMPLEMENTATION AND EXECUTION

Ref: (a) Sustainment Phase Configuration Management Process

Encl: (1) Configuration Management Process Discussion
(2) Configuration Management References and Resources

1. I am providing this letter to emphasize the importance of sound configuration management practices as a critical technique to reduce program risk and to remind Project engineers and Team leaders throughout the Marine Corps Systems Command of guidance available to help you in determining the optimum Configuration Management strategy for an individual program.

2. The Command is in the process of implementing a "Balanced Scorecard" as a means of measuring key characteristics of the work and human systems in the Marine Corps Systems Command. One of those key metrics is the "Systems Engineering Utilization Rate" which is broken down into five specific areas:

a. Have the system requirements been analyzed and verified from a technical perspective?

b. Are there formal technical reviews?

c. Are Technical Baselines under formal Configuration Control?

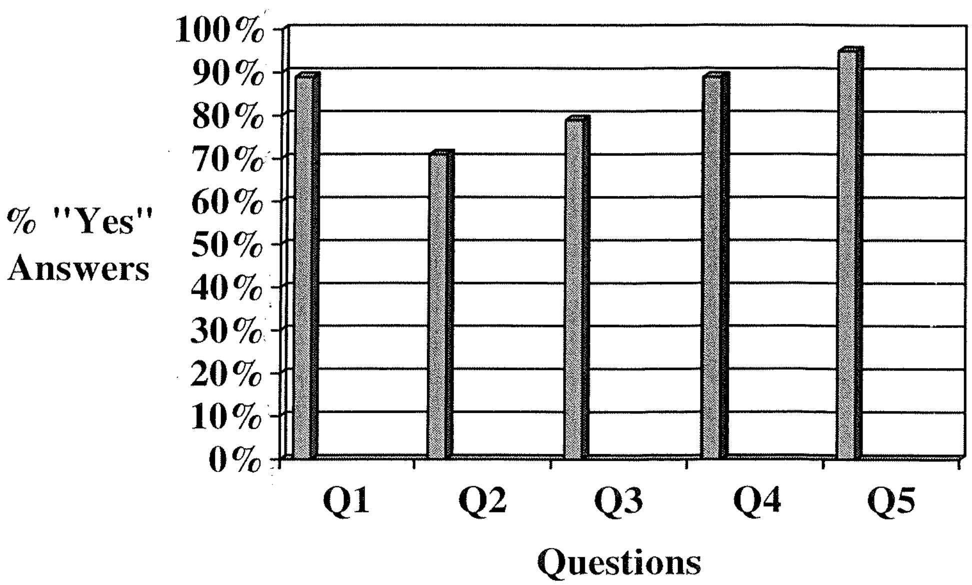
d. Is there a Requirements Traceability Matrix (RTM) or equivalent?

e. Have all system interfaces been identified and are they being properly managed?

Subj: CONFIGURATION MANAGEMENT-IMPLEMENTATION AND EXECUTION

3. We recently completed a review of selected programs and the results are summarized below:

a. As can readily be seen in this figure, of the five areas reviewed, the two worst (respectively) were the second (71% - Are there formal technical reviews?) and the third (79% - Are Technical Baselines under formal Configuration Control?). Both of these areas are directly related to the program's configuration management strategy and strongly suggest that we need to be more vigorous and disciplined in planning the configuration management portion of our acquisition strategies.



4. Program constraints, risk, schedule, cost, interfaces, and overarching strategy must all be carefully considered when determining the "best" Configuration Management strategy for an individual program. This in turn necessitates close coordination with the members of the Project Team, your Program Manager as well as the Lead Engineer for your Product Group.

5. The Configuration Management strategy for an individual program is reflected in the key program documentation such as the Marine Corps Single Acquisition Management Plan (MCSAMP). Enclosure (1) contains a broad discussion of configuration management issues and is intended to help program strategy teams

Subj: CONFIGURATION MANAGEMENT-IMPLEMENTATION AND EXECUTION

to determine the best configuration management approach; enclosure (2) contains a list of available configuration

management references and resources; and the reference contains a standard process applicable to that period of time in the life of a weapon system program between Full Operating Capability (FOC) and disposal, which is referred to in Post Production Systems Management as the Sustainment Phase.

R. L. Hobart
R. L. HOBART

Distribution:
Product Group Directors
Independently Assigned Program Managers
Assistant Commanders
Engineering Functional Integration Team Members

Configuration Management Process Discussion

1.0 -- Introduction

Configuration management -- the process -- is the same whether it is applied to hardware or software, weapon systems or information systems, development or sustainment. While each has unique requirements, the process used to document and maintain those requirements is the same. As shown in Figure 1, the primary purpose of the configuration management (CM) process is to ensure that the documentation used to identify and describe the product, i.e. the configuration baselines, and the product itself, are one and the same. Achieving this purpose ensures that we know the identity and capabilities of the systems we have developed and fielded and that they can be logistically supported.

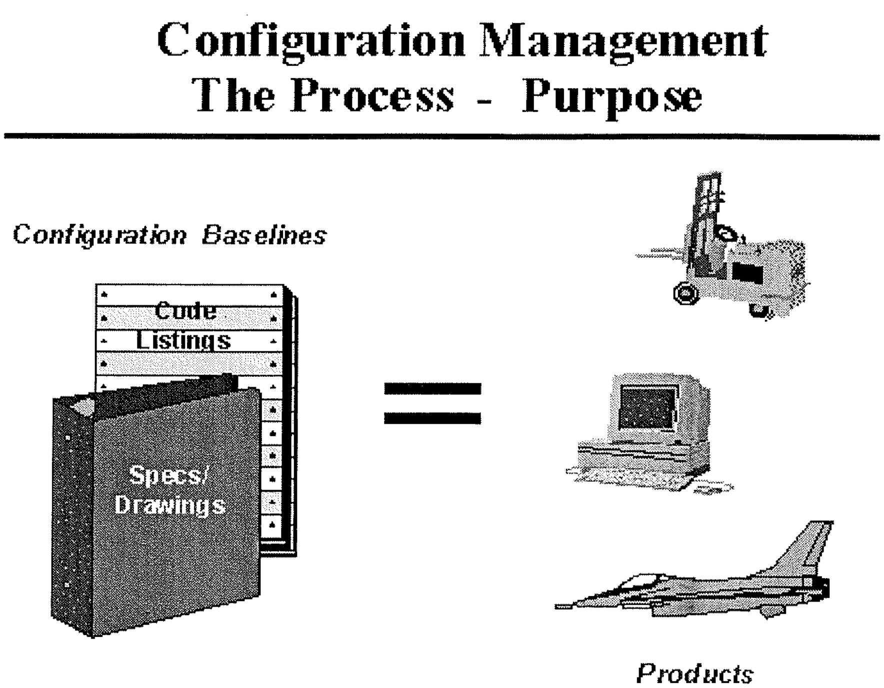


Figure 1 -- Configuration Management; The Process -- Purpose

2.0 -- Configuration Identification

CM is a subset of the systems engineering process. It's composed of a portion of systems engineering tasks which have been grouped together and assigned the title of CM. The purpose of configuration identification is to identify and document the decisions made by the systems engineering process regarding functional/performance requirements and physical characteristics of the system and its elements. This is accomplished by preparing and maintaining configuration baselines. These baselines are composed of specifications, drawings and code listings. There are three types of configuration baselines.

Functional Baseline: This baseline describes the functional/performance requirements and design constraints of the system as a whole entity. It is normally comprised of a system specification which defines system functional/performance requirements, system interface requirements, system technical constraints and all of the qualification provisions required to verify achievement of each specified requirement.

Allocated Baseline: This baseline describes the functional/performance requirements and design constraints of the system's elements. It is comprised of specifications which define the functional/performance requirements, interface requirements, and technical constraints for each system element. Also included are all of the qualification provisions required to verify achievement of each specified requirement.

Product Baseline: This baseline describes a verified design solution that meets the requirements and constraints of the allocated baseline. The product baseline expands the content of the allocated baseline specifications by incorporating engineering drawings and software code listings. It also incorporates requirements and verification methods needed for acceptance and first article inspections.

Acquisition reform uses performance-based specifications to the maximum extent feasible. Performance-based specifications describe requirements and boundary conditions such as interface requirements that a design must satisfy, not an actual design solution. From a CM perspective, the government would not take control of the verified design solution from the commercial developer using this approach, i.e. the government would not establish a product baseline. As shown in Figure 2, the functional and allocated baselines are performance-based; the product baseline is design-based. Using performance-based specifications for procurement means you could get a new design every time. This implies you would have to conduct a design verification (qualification) with every purchase, not a very cost effective scenario for weapon systems acquisition and support. However, conditions that exist within the commercial industrial base determine the feasibility of using a performance-based specification approach.

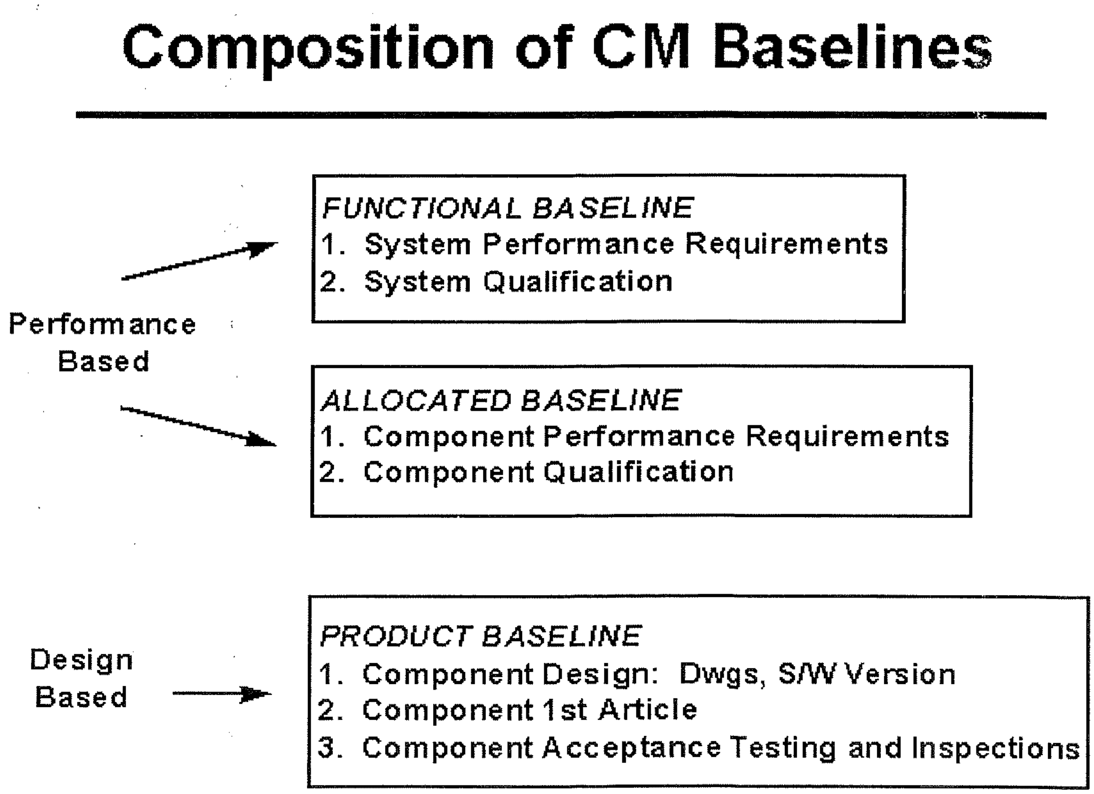


Figure 2 -- Composition of CM Baselines

Under most circumstances, we have to establish an allocated baseline so that we can define what we want a given element of the system to do. Only if we were procuring a turn-key system could we stop at the functional baseline. Once we know what the requirements are for the system elements, acquisition reform says that we should first determine if commercial products exist that can satisfy those needs. If they exist, we should use them in lieu of developing military unique parts. For example, if it's determined that you need 10W-30 weight motor oil for a ground vehicle, the commercial industrial base has many products that can satisfy the requirement. It's not necessary to identify a specific brand name or conduct a qualification program to verify that the product meets 10W-30 weight requirements because we have a high degree of confidence that the commercial industrial base has already verified the product. We don't need a product baseline that defines the chemical formula of the oil for the same reason. The same would be true if it was determined that we needed "D "cell batteries or a set of common metric socket wrenches.

For these simple examples, we would stop at the allocated baseline because we don't have to identify a specific design solution. We have been doing this for many electronic components all along. We don't get detail drawings for resistors, capacitors, etc. We don't qualify them to a resistor specification. We just specify that we need a 10 ohm resistor. Therefore, with a performance-based approach, we would purchase items using the allocated baseline instead of the product baseline. This approach should be used when we don't have to qualify a design or when qualification is very simple and inexpensive. This approach makes a lot of sense for government purchases that don't involve weapon systems. In most cases, we can't buy weapon systems commercially; we have to qualify military unique designs. Acquisition reform allows us to do this. However, we should use a performance-based approach to purchase as many weapon system components as feasible.

When more than performance-based specifications are required, it will be necessary for the government to take control of the verified design solution from the commercial developer by establishing and maintaining a product baseline. The level of design detail contained in a product baseline can vary greatly. It can be as simple as identifying a part number on a source control drawing or as complex as having detailed drawings for all of the individual pieces. While most of this discussion so far has used hardware examples, the same approach applies to software. For example, if after a requirements analysis, you determine a word processing capability in one of your sub-assemblies that includes the ability to bold, center and underline is needed, you've identified allocated baseline requirements. For the sake of discussion, let's also assume that only certain word processing software can satisfy the requirements. You can proceed along one of two paths. One says create new word processing software to meet these requirements. The other says purchase a commercially available product. Acquisition reform states the first choice is to buy a commercial product. Let's say you choose Microsoft Word. After verifying that Word meets the requirements, you would incorporate the design solution into the product baseline by identifying Microsoft Word, Version "X", as a component of the sub-assembly. From a CM perspective, the level of detail is only at the very top level, i.e. buy this version of this specific software. We would not identify the actual code used in Version "X". If we had chosen the other path, our product baseline would have incorporated a listing of the software code into the product baseline.

Another good example to help explain the difference between the allocated and the product baseline is the KC-10. The government purchased a commercial product, the DC-10, as a key component of the product baseline. Even though it was modified, the DC-10 was a specific design solution that met the air vehicle performance requirements identified in the allocated baseline. If you implemented a performance-based approach, i.e. bought to the allocated baseline, a prospective bidder could have use an L-1011 for one buy, a 747 for another or a DC-10 for a third, as long as it met the air vehicle performance requirements. Therefore, when it's necessary to identify a specific design solution, be it government unique, a commercial product, or a modified commercial product like the DC-10 in this example, it's necessary to establish a product baseline.

The program's maintenance philosophy should determine the level of detail documented in the product baseline. If you are going to buy spare parts, you need documentation that lets you purchase them as separate items. If the maintenance philosophy says replace the box, you need documentation down to the box level. If the maintenance philosophy calls for replacing circuit cards in the box, you need more detailed documentation that lets you purchase spare circuit cards.

Product baseline documentation can be in the form of specifications, drawings or code listings. The kind of documentation needed depends on the product. If it's a simple metal part, you may only need a drawing. If it's a higher assembly, you may need specifications, drawings, and code listings, if there's software in the assembly.

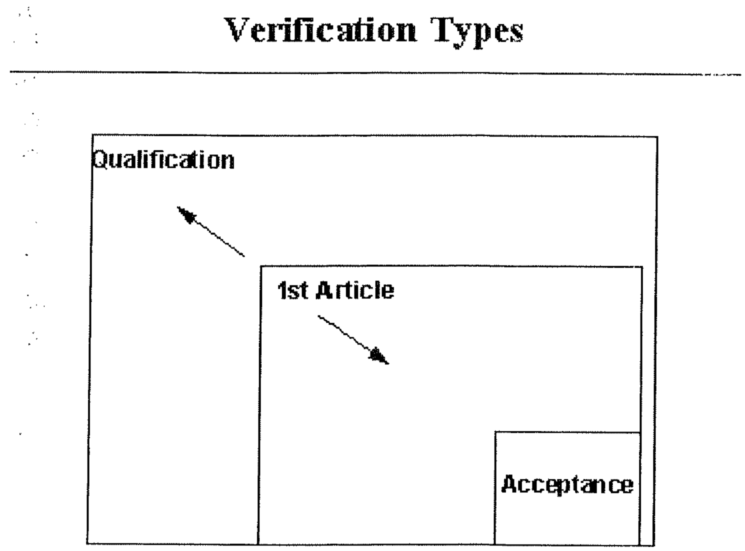


Figure 3 -- Verification Types

The final part of the configuration baselines are the types of verification. There are three main reasons for conducting verifications. First, you want to verify that a new design or modification meets its requirements. Second, you want to verify that a manufacturer can build a design that has already been verified. Third, you want to check that items coming off the assembly line are working properly and are ready for shipment. The first we call qualification, the second is first article and the third is acceptance (See Figure 3).

Qualification is the most complex type of verification. For example, it's usually conducted at the temperature extremes. Acceptance is the least complex, often conducted at ambient conditions. First article usually falls somewhere in between the two. It can be as complex as qualification or almost like acceptance depending on the design. Qualification methods are defined in the functional and allocated baselines. First article and acceptance methods are design dependent; they are defined in the product baseline. Therefore, if you implement a true performance-based approach, i.e. stopping at the allocated baseline, you can't have first article or acceptance types of verification.

3.0 -- Configuration Control

The configuration control board (CCB) is still the only body that establishes and approves changes to the configuration baselines. Engineering change proposals (ECPs) are still the only vehicles that authorize changes to those baselines. The key here is that we only submit ECPs against what we control. If we only have control at the top level such as identifying a manufacturer's part number on a source control drawing, we would only process an ECP if we needed to change to a new part. Changes the manufacturer makes to internal components are not submitted to the government for approval because we don't have control to that level of detail. This scenario holds true for requests to depart from the baselines as well. We would only process waivers and deviations against what we control.

What acquisition reform is asking us to do is to elevate our control level, leaving the lower detailed levels under the manufacturer's control. How much control we need is a risk management issue. What happens if a manufacturer makes a design change? How do we know? Is he obligated to tell us? Do we have to re-verify the design? Electronic components probably experience as much change as any product on the market today. If the manufacturer changes to a faster microprocessor, will the box that we purchased from him in the past still work with other components in our weapon system? Keys to determining the risk involved is knowing how well the commercial industrial base understands the intended use of the product and how they factor those kind of needs into their design process. When the personal computer world decides to change to a faster chip, they factor in the knowledge that millions of customers have millions of dollars worth of software designed for slower chips. Those customers are not going to buy all new software just so they can have a faster computer. Therefore, in order to sell faster computers, they design the new computers to run the old software. In essence, they know their customer base and their needs when designing. As we try to use more commercial products in our weapon systems, is the manufacturer figuring in military needs when designing? Does he understand our environments and uses? To reduce risks before buying a commercial product, we should ask the manufacturer if he has made any design changes since the last time we purchased his product. Based on the information provided, we can determine if a product re-verification is necessary before proceeding with the purchase.

4.0 - Configuration Audits

Configuration audits are conducted to verify that a proposed design solution meets the specified requirements and that documentation describing the product is accurate. Functional configuration audits (FCAs) are done at the end of EMD to ensure the proposed design solution satisfies the performance requirements of the system and development specifications and that we know the configuration of the test articles. Physical configuration audits (PCAs) are conducted on the first representative production unit to ensure that the verified design solution documentation is accurate and that its acceptance procedures are valid.

Acquisition reform doesn't change how we conduct audits but it can eliminate the need for a PCA. Like the change control process, we only audit that which we plan to control in the configuration baselines. If we were buying black boxes, we may just have a source control drawing with the part number on it. A PCA would make sure that the proper part number was identified on the drawing but we wouldn't audit drawings of the box components. Similarly, if we purchase a commercial software package, we would make sure the name and version number were correctly documented, but we wouldn't audit the code listing. However, with a performance-based approach, there wouldn't be a product baseline because we wouldn't identify a specific design solution. There wouldn't be a source control drawing for the black box or a specific software package identified. Therefore, there wouldn't be a reason to conduct a PCA.

5.0 - Configuration Status Accounting

Configuration Status accounting serves as an information library. We refer to this information as the technical baseline. The technical baseline is composed of all of the technical information needed to support a system throughout its life-cycle. We divide this data into two categories -- configuration baselines and decision support data (see Figure 4). Configuration baselines, as discussed earlier, is data that must be processed through a CCB for approval. All other technical data is referred to as decision support data. Some decision support data requires approval but it is not processed through a CCB. Examples are TMs or test reports.

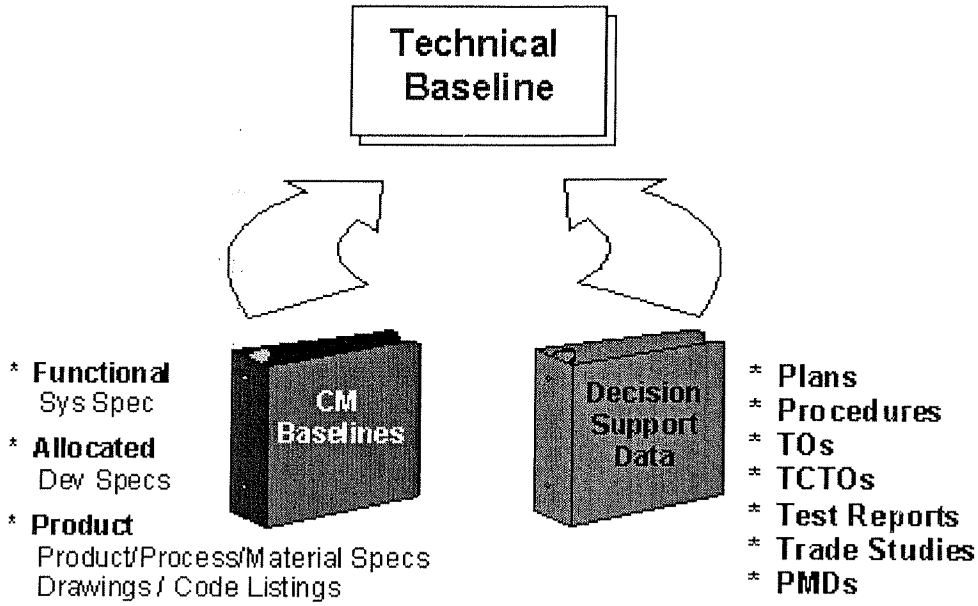


Figure 4 -- Technical Baseline

6. Program's Dilemma -- The Optimum Mix

In the final analysis, there isn't a standard template for implementing CM. Each program must determine an optimum approach by comparing program requirements and available government resources to the commercial industrial base capabilities. Organic repair commits the government to a logistics support enterprise requiring personnel, training, facilities, equipment, procedures, spare parts, and documentation. If these resources are not available, organic repair is not a viable option. Faced with dwindling resources, programs must chose the most cost effective maintenance strategy -- organic, commercial or some combination of the two. If we establish a product baseline with just a manufacturer's part number identified and want competitive re-procurements, then we need to qualify a second design so that we can compete one part number against the other. On the other hand, if we want to support a single configuration competitively, we need to establish a product baseline that documents the verified detailed design. Correctly implementing the systems engineering process will lead to a solution that balances the best mix of commercial and military unique products.

Configuration Management References and Resources

Mil-Hdbk-61, Configuration Management Guidance

This military handbook provides guidance and information to DoD acquisition managers, logistics managers, and other individuals assigned responsibility for Configuration Management. Its purpose is to assist them in planning for and implementing effective DoD configuration management activities and practices during all life cycle phases of defense systems and configuration items. It supports acquisition based on performance specifications, and the use of industry standards and methods to the greatest practicable extent.

EIA 649, National Consensus Standard for Configuration Management

"National Consensus Standard for Configuration Management" as the guiding document providing the basic principles of Configuration Management. DoD has been instrumental in the on-going development of EIA-836, "Consensus Standard for CM Data Exchange and Interoperability" and will adopt it when it is published by the Electronics Industries Alliance as a web-based asset. This limited coordination revision to MIL-HDBK-61 is being issued to provide continuing up-to-date guidance for effective application of configuration management as the transition from MIL-STDs continues.

DoN Turbostreamliner

Turbo Streamliner is a repository of acquisition definitions, principles, best practices, lessons learned, references, sample contractual language and related web sites. This tool is organized by acquisition functional areas and associated Request for Proposal (RFP) elements. The purpose of Turbo Streamliner is to assist the acquisition community in applying acquisition reform (AR) principles, concepts and techniques to acquiring and sustaining Department of the Navy (DoN) warfighting capabilities for new requirements and reprocurements. The section that addresses Configuration Management can be found at http://www.acq-ref.navy.mil/tools/turbo/topics/w.cfm

CM Navigator

The Configuration Management Navigator is a tool to assist Program Team members in applying Configuration Management (CM) on their program(s). One objective of this tool is to provide an alternative to a traditional handbook; using a road map approach showing the significant CM activities and products in relation to the acquisition phases. Explanations and how-to procedures for these activities/products are contained in series of text and graphic files that are hyperlinked to the CM road map. Available on the PM Tool Kit in TIGER.