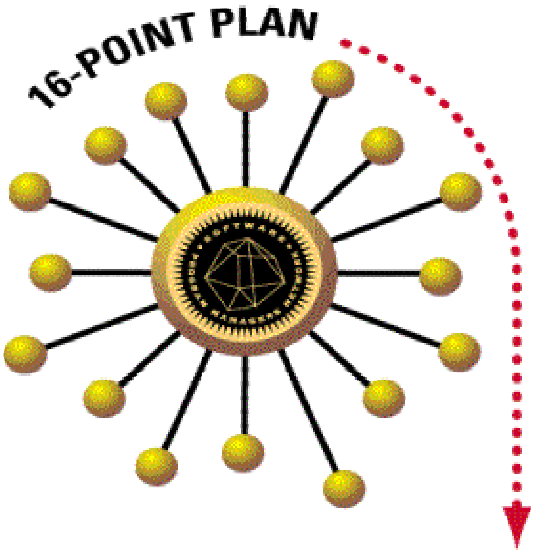
American Systems Corporation
Employee Owned…A Better Company




# 16 Point Plan™ Configuration Management Guidance

***PRACTICE 7; OVERVIEW & APPLICATION GUIDANCE***

## Little Gray Book

Copyright © 2002 American Systems Corporation. All Rights Reserved.

**WARNING** - This document includes data that shall not be duplicated, - in whole or in part without prior written approval. The information contained herein may be used and applied to specific program and organizational requirements to conduct the specific tasks described herein. This restriction does not limit the recipient's right to use information contained in this data if it is obtained from another source without restriction. The information subject to this restriction is contained in sheets marked accordingly.

# TABLE OF CONTENTS

PREFACE ..... v
CHAPTER 1: INTRODUCTION ..... 6
The Configuration Management Requirement ..... 7
Project Impacts of Inadequate CM ..... 8
Organizational Roles and Responsibilities ..... 9
The Government's Role In CM ..... 10
The Contractor's Role In CM ..... 10
CHAPTER 2: THE PROGRAM ENVIRONMENT TO BE SUPPORTED BY CM ..... 12
Documentation Relationship ..... 13
The Life Cycle Relationship ..... 15
Life Cycle Documentation ..... 16
Planning Documentation ..... 18
Technical Documentation ..... 18
Management and Technical Reports ..... 19
Product Assurance Documentation ..... 20
The Documentation Relationships ..... 20
The Life Cycle Relationship to Engineering ..... 21
Government's Role in the System Life Cycle ..... 22
The Contractor's Role in the System Life Cycle ..... 22
CHAPTER 3: CONFIGURATION IDENTIFICATION ..... 24
The Transition Process ..... 24
Engineering Transitions ..... 26
Partitioning of CM Information ..... 28
Transfer of Data ..... 29
Configuration Control ..... 29
Controlled Information Area ..... 31
Configuration Management Area ..... 32
Documentation Area ..... 32
Test Area ..... 32
Government Area ..... 33
Configuration Identification and Formal Baselines ..... 33
Functional Baseline ..... 34
Allocated Baseline ..... 34
Product Baseline ..... 35
Additional Baselines ..... 35
Baseline Structure ..... 36
The Role of CM in Identifying and Controlling Baselines ..... 36
Organizational Roles and Responsibilities ..... 39
The Government's Role in Configuration Identification ..... 39
The Contractor's Role in Configuration Identification ..... 40
CHAPTER 4: CHANGE CONTROL ..... 41
The Change Control Process ..... 41
Types of Changes ..... 42
Class I Changes ..... 42
Class II Changes ..... 43
Internal Changes ..... 43
Change Control Boards ..... 44

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation

Use or disclosure of this information is subject to the restriction on the title page of this document.

Control of Change to Non-Baselined Information ....44
Change Control to Baselined Information ....45
Configuration Control Application ....46
System Problem Reports (SPRs) ....47
Change Board Processing ....49
Interface Changes ....50
Deviations and Waivers ....51
Change Closure ....51
Change Control Tailoring ....52
Organizational Roles and Responsibilities ....52
The Government's Role in Change Control ....53
The Contractor's Role in Change Control ....53
CHAPTER 5: STATUS ACCOUNTING, REVIEWS, AND AUDITS ....55
Status Accounting ....55
Version Description Documentation ....56
Subsystem Status Summary ....56
Baseline Correspondence Report ....56
Configuration Index ....56
Status Report ....57
Maintenance of Baseline Status Information ....57
Reviews and Audits ....57
CM Audits ....58
Baseline Reviews ....59
Acceptance Reviews ....60
Organizational Roles and Responsibilities ....60
The Government's Role in Status Accounting ....60
The Contractor's Role in Status Accounting ....61
CHAPTER 6: CM ORGANIZATION AND PLANNING ....62
CM Planning Requirements and Relationships ....62
System CM Planning ....64
Software CM Planning ....66
Software Configuration Management Plan ....68
Additional Software CM Planning Factors ....69
Organizational Interfaces ....70
Configuration Management ....70
Project Characteristics ....71
Information Access ....72
Final Considerations ....73
Government's Role in Planning ....73
Contractor's Role in Planning ....74
CHAPTER 7: SAMPLE CM APPLICATION CONSIDERATIONS ....75
The Data Explosion ....75
Analysis of “Data Explosion” Problem ....77
Potential Solution to the “Data Explosion” Problem ....78
The Distributed CM Environment ....79
Analysis of the Distributed CM Problem ....79
Potential Solutions ....80
CM Control of Large Amounts of Information ....80
Analysis of the Size Problem ....81
Potential Solutions ....82
Control of Object-Oriented and Ada Projects ....82
Analysis of the Ada CM Problem ....84

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

Parallel Development....84
Parallel Development Analysis....85
The Government's Role in Preventing CM Problems....85
The Contractor's Role in Preventing CM Problems....86

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

## PREFACE

Since completion of the contract held by the ICE directorate of ASC to support the Software Program Managers Network (SPMN) and the Best Practices Initiative ASC has continued to extend and maintain these practices and disseminate them through training and maintenance of related Web sites. The corporation recognizes the critical nature of this support to the DoD software community and has continued this support at their own expense.

Please visit our Web sites, www.2asc.com, www.iceincusa.com or www.spmn.com . Any comments or requests for further information should be made to Mike Evans at Michael.Evans@2asc.com

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com


WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

# CHAPTER 1: INTRODUCTION

This handbook on ***Configuration Management*** (CM) addresses the poorly understood and often critically mismanaged CM process.

The CM process generally consists of the following important functions: identification of information used or developed by a project; **management of change** to information where the changes impact organizations, management practices, schedules, and budgets, technical or assurance activities or project status; information ownership where the ownership is critical to the integrity of the project and the organizational agreements which have been made; project delivery and release management procedures and the capability to monitor the status of project information; and project configuration reviews and audits which assess the status and acceptability of products controlled by CM or released.

This handbook is a practical guide on how to plan, implement, manage, and control the process of CM. While consistent with the DOD Standard 973, the CM standard, it is based on actual project experience. The handbook provides "hands-on" guidance, which is intended to demonstrate to a project *what to do* and *why it is important*, not how to implement the procedure. To present an integrated view of information control, the approach combines CM and related engineering disciplines. These disciplines include:

- Product assurance
- Engineering management and application
- Walkthroughs
- Audits
- Inspections and reviews
- The role of the Computer Aided Software Engineering (CASE) environment in supporting the CM disciplines.

This departure from the normal view of CM as “just another administrative function” demonstrates the centralized and critical information management function provided by CM.

This handbook is particularly useful for managers responsible for software development and support of software; personnel concerned with planning, controlling, or working within a software project requiring CM; CM personnel responsible for planning or implementing an information management and control environment; and quality personnel responsible for monitoring and evaluating the engineering environment and products.

The handbook is also useful for customer, project, quality assurance, technical and management personnel who must interface with software CM in a project situation.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

## *The Configuration Management Requirement*

"*What the heck is all this administrative garbage for? Why can't I just be left alone to do what I do best—program computers...?"*
This quote from an anonymous programmer represents the attitude of many software engineers and, unfortunately, many engineering managers who control and direct their efforts. Despite many examples of software projects failing due to uncontrolled changes and/or poor management of essential information, the need to rigorously manage information in a project environment is still not always recognized.

CM is the means by which shared information, whether it is produced, used by, or released from a software development or support activity, is controlled and maintained.

CM methods provide a means to identify, track, and control system development from the inception of the concept for the system until it is replaced or retired. Management of baselines and engineering products through CM provides a sustained control of the information as the engineering functions work their way through the process.

Application of CM to a project environment provides the discipline needed to ensure the system operational concept is consistent with the capability that results. Based on the criticality of the system, the rigor associated with CM and the evaluation of information system will vary.

The fundamental purpose of CM is to aid in the planning of, and control the changes to, a software configuration. CM is an integration of formal control techniques that maintain the integrity of approved information and data. The CM disciplines manage the validity of contractual relationships, data management and control practices, tools and project facilities.

CM applies technical and administrative direction and surveillance to:

- *Identify and document* the functional and physical characteristics of a system, software, hardware, or operational component such that these relationships may be maintained, controlled ,and assured.
- *Control changes* to those characteristics.
- *Record and report* configuration change processing and implementation status.
- *Review and audit* the CM process to assure the process and the adequacy of control.

The goal of these operations is to assure the delivered configuration item meets "form," "fit," and "functional" requirements. In simple terms, CM provides: the ability to identify what information has been approved for concurrent use in the project, who owns the information, how the information was approved for CM control, and what is the latest approved release.

It also provides a means to assure that before a change can be made to a piece of information for which you are the responsible person, or that impacts work that you are

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

doing, you have the option of approving the change or providing input into the change approval process.

Through CM, the project provides an accessible and current record of the status of each controlled piece of information that you might plan on using, the content of each release made from CM and who has checked out or is working on a piece of information that you plan on accessing through CM. The exact content, status and version of all information accepted by CM is available, and when it was last accepted and when it was last updated.

CM is an essential layer in the overall software environment used to produce and/or maintain a software product. If CM is not included as an integral part of the software environment, the project's end results may be catastrophic.

### *Project Impacts of Inadequate CM*

Programming is often perceived as a highly individualistic activity practiced by talented, brilliant loners. True programming does, in fact, have many of those characteristics. An individual has an idea, designs a software solution, codes and tests it, and produces or enhances an application.

Software engineering, on the other hand, is a different engineering problem altogether. In this case, a need to provide or augment an operational capability is identified. In response to this need, specific requirements are identified and a system engineering project is specified, funded, and initiated. The software component is specified to satisfy a specific set of requirements, then the effort is funded, scheduled, and work is initiated.

The differences between the software engineering requirement and the activities of a programmer are the following:

- The software engineering effort must satisfy a predetermined set of requirements rather than an individual programmer's conceptual operational requirement.

- Software engineering requires the coordination of many diverse disciplines and different individuals in order to develop a product that meets a predetermined capability.

- The software engineering process requires the integration of many different methods that interact through the information which each method produces.

- The individual engineering products must be linked together with traceability maintained throughout the engineering information elements.

The disciplines surrounding programming do not normally emphasize information management and control. The development is normally conducted by a single programmer or a small team. The engineering effort is localized to a single room. Because of this, whenever a change is required, the change can be discussed and

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

applied with the complete cognizance, agreement and knowledge of the programming team.

Software engineering applications must provide an environment that enables communication and assures a smooth controlled flow of information. Many different organizations, often distributed across rooms, floors, buildings, or sites, must work together to satisfy a requirement within a predetermined cost and schedule. Experimentation, rework due to poor control of information, and uncontrolled change cannot be allowed if the engineering effort is to succeed. The control of information content and quality, the control of releases, and the management of change to released information is critical to project success.

Symptoms of inadequate configuration management are not always obvious, they are usually difficult to address and easy to misdiagnose. Problems such as non-traceability, interfaces and operational requirements that are poorly satisfied, or components which will not integrate together because of interface inconsistencies are blamed on technical rather than information management problems.

In many instances these catastrophic engineering problems occur because of inadequate information management and control—the problem solved by CM. The CM disciplines are one of the prime risk reducers in a project environment.

### *Organizational Roles and Responsibilities*

The CM process provides the structure through which the customer and the contractor fulfill their CM roles and communicate with each other in a disciplined manner. Both the contractor and the customer share a common goal—usually defined as delivery of the specified product on schedule and within budget.

CM offers a disciplined process to both organizations through which the following information, critical for well-managed programs, is made available:

- Product identity
- Product location
- Product status
- Product change history
- Product relationships

Without CM, the contractor could never deliver a quality product. The development organizations would be in a constant state of confusion. It would be impossible to identify the most current version of a product, relate products to one another, or identify changes to a product. Likewise, the customer would have little insight into product development, schedules, change status, or the disposition of review comments.

A strong and vigorous CM program requires the commitment of the contractor/customer team from the beginning of the acquisition phase until the system is retired.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

### *The Government's Role In CM*

The list below indicates those things that the government is responsible for in a successful CM process. The government's role is to:

- Establish a government CM organization that is responsible for approval and acceptance of deliverable products. This organization may control the release of the information to the operational environment.
- Specify CM policies for all related contracts to ensure sufficient commonality of implementation of CM processes. For example, they may require that all contractors establish a common CM framework, such as, Change Control Boards that meet weekly, with meeting reports generated, reported on, and available for audit.
- Specify CM standards for all related contracts to ensure sufficient commonality in format and content of all deliverable products, and to ensure a common interface to customer CM processes.
- Establish criteria for acceptance of deliverable products that directly address CM requirements.
- Include other appropriate requirements for CM in related contracts and Statements of Work (SOWs).
- Ensure sufficient contract funding to support the CM function.
- Participate in the development of CM procedures and the selection of CM tools.
- Participate in reviews and audits.
- Support Change Control Boards.

### *The Contractor's Role In CM*

The contractor is responsible for the main part of the Configuration Management duties. The contractor must:

- Define comprehensive CM plans and procedures that comply with all customer requirements.
- Implement and document CM procedures.
- Build and maintain the CM environment.
- Manage the review and change process.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- Deliver a complete and consistent set of program products.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

## CHAPTER 2: THE PROGRAM ENVIRONMENT TO BE SUPPORTED BY CM

While there are several critical relationships that are essential to the success of a software project, probably the most critical is the relationship between the engineering organizations and the organizations which interact with or use the products which are produced. The organization sponsoring the project is responsible for defining what is needed for a system (or portion thereof) and for procuring it, assuring that what is delivered is satisfactory, and integrating it into the system that is produced for operational use. This is usually done by the system organization interfacing the end users and engineering staffs.

The engineering organization is responsible for providing the hardware and software component as required by the sponsor and specified by the system organization. The software organization is normally responsible for engineering the software, but any organization can perform the engineering function as long as the responsibility is delegated, controlled, and there is an organizational interface structure established between the various groups.

The project life cycle is the basic structure for coordinating the interaction between the engineering organizations and external organizations that use or support the application. This structure establishes an integrated way of doing business that can be applied to all program organizational components.

A life cycle consists of a series of steps, called phases. Each phase includes a set of activities and the products from the activities, including documentation. Assurance activities, including formal testing and reviews, are specified to ensure the adequacy of products and activities and to indicate readiness to proceed to the next phase. A life cycle is nothing more than a interface standard between diverse segments of a project. The various life cycle levels—acquisition, system, hardware, software, and operational procedure development—ensure that the policies and practices used to develop, modify, assure, organize, control and release information are consistent within and across project areas. The life cycle ensures that each specific engineering application will require only the rigor and complexity that is consistent with the characteristics and application requirements of the project, and that nonessential activities, documentation, and reviews are not required.

A life cycle should provide:

- Planning and engineering controls and data that will enhance predictability and lower risk when developing, maintaining, or supporting systems and software.

- A time-phased order in which documentation is to be written, and how this structure is to be adapted to various project environments.

- Guidelines for determining the amount, form, structure, and relationships that exist in documentation that is to be applied to a particular project environment.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- An integrated project environment for applying CM and control and assurance of systems, software, and other engineering and operational interfaces and products.

- A means to expand the minimum documentation set into multiple documents, while ensuring traceability in order to reflect the complexity of the project, type of engineering effort, or physical project characteristics.

A program life cycle is, in reality, an integration of several life cycles tied together through baselines established when specific events occur and approvals are given. Information relationships that exist between life cycles are a mirror image of each other. For example, the contract constraints from the system life cycle shape the planning requirements; the system interface, performance, and operational requirements feed the software concept; and the system design feeds the software requirements.

Two main types of life cycles are system life cycles and software life cycles. A system life cycle structures the acquisition of the software component. The software life cycle and its associated documentation and review requirements establish the basic context within which the software is built. The software life cycle must provide a framework to address documentation and engineering relationships for software integrated with the system requirements. Each phase of the software life cycle ends with the occurrence of a milestone event and produces identifiable documents or software items, or a combination of both.

### *Documentation Relationship*

The disciplines surrounding CM have evolved from the systems engineering and management procedures used to engineer products during the late 1960s, '70s, and '80s. Throughout this period, frequent problems were experienced in the engineering of large- and small-scale projects. The complexity of the engineering, the criticality of the operational environment that the product was to be applied to, and the amount of information that had to be managed to deliver, document, and maintain the product forced the development of an information management discipline to control and manage information baselines.

The CM disciplines maintain the information relationships between organizations, engineering levels, information components, and between the sponsor of an engineering activity and the engineering group building or supporting the product. The common link between each of these relationships is the form, structure, content of data, and information developed or used by a project.

Three information-related factors shape the choice and application of a CM strategy to a particular project environment:

- The information framework within which all information used, developed, or modified by a project determines the structure, relationships, and data and information interfaces that must be maintained by the CM environment.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com


WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- The types and amounts of information that must be managed are essential to consider when adapting CM disciplines to the project environment.

- The physical characteristics of the project environment are the basis for the organizational, administrative, and procedural requirements of the CM environment.

The information framework is the guidelines used or developed by a project in support of the management, engineering, assurance, or life cycle documentation. The framework provides the means to accomplish the following:

- Determine the specific planning, technical, assurance, and report information that must be developed and documented to plan and describe the software.

- Determine the order in which documents are to be written.

- Determine the amount of information to be formally recorded.

- Provide a basis for CM and control and assurance.

This document structure ensures that, for each life cycle component, essential activities are conducted and critical required information is captured and retained. The software engineering process is, in a large measure, the production and management of information. The total cost of producing, managing and updating documentation of all forms if a project is 60 percent of the overall cost of the project.

The most common form of information that must be managed through CM are the formal, baselined documents that are approved by the sponsor of the engineering group, and establishing the contractual, or formal, relationship between the sponsor and the engineering group. These formal documents control the acquisition of the software by a separate organization.

The formal documentation, although essential to define and document management and technical agreements between engineering and acquirer, are not the means by which the engineering process is described or documented.

The engineering strategy produces small pieces of data which are aggregated into different models describing specific views of the engineering and engineering management process. Each of these separate classes of information is included in the information framework and is managed through different, but related, segments of the CM process.

The framework includes all formal documentation, all shared engineering information, and all data products that are locally used by an engineering method or organization. The CM disciplines described in the following sections provide the environment to manage the integrity of all formal data and all planning, management, and engineering information and data products shared in a project environment.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

## *The Life Cycle Relationship*

A system is an operational capability comprising software, hardware, and operational and support components. Software is an essential component, but is only a component in the system configuration. The system engineering process provides a direct interface between user and operational environment. The system life cycle, and its associated documentation and review requirements, establish the basic context within which smaller elements of the system; subsystems; hardware, software, and operational components of each subsystem; and the smallest pieces, processes, and units are acquired, fabricated, or developed. The system life cycle must provide a framework to address documentation and engineering relationships and have sufficient flexibility to facilitate project tailoring.

A life cycle looks at the process of engineering as a serial occurrence of discrete engineering activities. These activities are linked through the formal specifications that describe the results of the particular phase of development.

Under this approach, specifications are developed as formal documents at discrete points in the development cycle, approved by the user or buyer of the system, and from that point forward, they are controlled as an integrated set of project baselines. This view of engineering assumes that the process is rigorous and flows smoothly from one phase of development to the next.

A life cycle is also the means by which a recipient of a system capability maintains visibility into how the operational capability is defined, designed, acquired, and released. System components are not directly fabricated under this life cycle. Life cycles do, however, provide the ability to coordinate developing various components and operational procedures through lower level life cycles. The links between them are the formal documents that are produced by the life cycle, the review and approval procedures that result in release of information from the life cycle, and CM procedures that control data at each life cycle level. This CM life cycle is graphically detailed in Figure 2.

Between system and component life cycle levels, the design from one life cycle forms the basis for the lower level life cycle requirements. Detailed interface specifications at one level structure the lower level interface requirements; procedures from the higher level life cycle form the basic management structure and requirements for the life cycle below; the work package definition at the level above provides the basic task requirement for the life cycle below; the production schedules at the level above are the delivery requirements for the life cycle below; the higher level budgets are the funding profile for the lower level life cycle; and, finally, the productivity requirements at the higher level are decomposed, allocated to the component life cycles and used as the productivity measures for the engineering levels below. CM must provide the consistent and controlled project framework to manage the information relationships between life cycle levels.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

At the highest life cycle level, CM deals with elements known as Configuration Items (CIs). A CI is defined as a system or subsystem component that, for technical or management reasons, is contracted for, or produced as, an entity.

There may be multiple CIs in a system; each being tracked through CM. For example, a CI may be as simple as the software, documentation, and operational procedures that provide a personal computer with an enhanced spread-sheet capability or as complex as a complete data processing capability composed of facilities, electronic computers, interfacing equipment, software, and documentation.

Under a CI are Hardware Configuration Items (HWCIs) and Computer Software Configuration Items (CSCIs). Hardware CM functions are not described in this text, although there are specific common relationships that exist between the software CM functions and those used for hardware.

The CSCIs are the ordered set of instructions and documentation that, in aggregate, fully describes the requirements, design, and technical content of a software component. ACSCI is allocated from the system design requirements and, when completed, is integrated into the system as an entity.

The next lower level that is managed through software CM is the Computer Software Component (CSC). The CSC is normally a loadable entity that may be decomposed into lower level CSCs or Computer Software Units (CSUs). A CSU is the lowest-level software entity that is tracked and managed through CM. A CSU is normally less than 10 subroutines and 1,000 lines of source code.

The CIs, CSCIs, and HWCIs are normally end items that are delivered, with their documentation and supporting products, to the organization sponsoring the engineering activity. As such, the CM process must include the formal control practices to coordinate the contractual, management, and technical interfaces. This management and control discipline is two-way; each side of the life cycle interface must provide and support a CM and change management discipline that ensures the consistency and integrity of agreed-to information. Lower-level CSCs and CSUs are controlled by the engineering organization's CM, as they deal with information not approved by the sponsor but required by the project. This control level and discipline manages the information relationships between technical and organizational entities and assures that changes to shared project information is applied in a controlled manner.

### *Life Cycle Documentation*

The formal documentation is the window through which all organizations interfacing to a software project view the management, the technical process, and the progress towards completion. It is as essential to the software engineering process as are the methods and techniques that manage, control, assure, and engineer the product.

Through the formal documentation, the sponsor of a software engineering activity may specify what is wanted, what operational system and subsystem interfaces must be

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

supported, and the resources that have been allocated by the sponsor to engineer the product.

The documentation structure provides a framework for packaging engineering information in a form that may be reviewed and evaluated by an external organization. As such, the integrity of documentation and maintenance of critical relationships between documentation segments is essential to project success. Managing the change status of all information that impacts different organizations and users of the documents is critical to the success of the project and the acceptability of the products that result. CM provides these management functions.

Documentation becomes the formal record of the technical approach, schedule, and cost that is agreed to by management and the developer at the completion of each review. It provides a common foundation for continued development by different organizational units or by different contractors working on the same project.

Certain documents may be prepared at the system life cycle level, rather than separately at the software component level. The documents for Certification, Acceptance Criteria, Government Inspections, and Acceptance Reports, for example, would be prepared at the system level rather than the software level.

How much documentation does a program need and how formal must the control be? If the software is produced by four programmers in a room, the formal documentation may be minimal; although the engineering documentation may be required, communication requirements are satisfied by proximity. If two of the four are widely distanced from the others, and the application is embedded in a complex system, engineering information must be converted into a set of formal documents to facilitate review, approval, and information control.

The information base is fixed for a project and must be defined at the beginning of the engineering effort. This base must be produced by the software project if the engineering risk is to be acceptable, and the results of the project are to be predictable, maintainable, and acceptable.

The number of documents that this base is mapped into and the specific outlines that are used to include the individual components are less important than the fact that the base is complete, consistent with the characteristics of the project, and actually developed as part of the engineering, management, or assurance process. Furthermore, it is not necessarily helpful to issue a separate document for every individual documentation format unless required by software system size or complexity.

The term “documentation” should be interpreted as meaning that the information may be covered as a separate document or as a section within a larger document as is most appropriate to the size and complexity of the software being developed for a particular system.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

The formal documentation structure is an essential element of the life cycle structure, and it is the means by which information is passed between life cycle levels. Formal software documentation can be grouped into four distinct types:

- Planning Documentation—Describes how development of the system is to be managed, controlled, and monitored. These documents identify the resources and constraints to be applied to the development effort.

- Technical documentation—Describes the technical aspects of the system at each life cycle phase and structures the engineering models in a formal document.

- Management And Technical Reports—Document the development or support experience and describe the significant milestones and activities that occur.

- Product Assurance Documentation- - Documents the status of the products that are developed or acquired and the integrity of the processes used to produce them.

### *Planning Documentation*

Planning documentation represents an agreement between the sponsor of an engineering activity and the engineering organization as to how the software and its component parts are to be produced, the resources to be applied, a plan for application, and the constraints that must be adhered to during the development. At each planning level (within a life cycle and across the life cycle hierarchy), each individual plan should include:

- The definition and allocation of work
- Schedules
- Budgets
- Detailed methods and approaches for management
- Plans for resource acquisition and assurance.

The specific plans at each level must conform to any higher-level plans or policies.

This management documentation, which spans organizations, program life cycles, or phases within a life cycle, should be managed by CM just as any other piece of information. Changes to the documentation must be approved by the owner of the plans. Uncontrolled distribution, access, or poor version management will adversely impact the integrity of the project.

CM provides the means to assure consistency of the planning documentation and assure that changes to plans are coordinated with all organizations affected by the change and approved by the plan owner prior to implementation.

### *Technical Documentation*

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

There is, at each life cycle level, technical documentation that describes the product being produced by that life cycle component. This technical documentation records the information related to the engineering and integration process (for example, development) of the software or component. The documents are normally the responsibility of the engineering organization. The specific documentation requirement for the life cycle level (for example, number, content, and format of separate documents) is defined in the development plan for that level.

The technical documentation requirement is specific to a level of the life cycle. Although the basic documentation structure is common for each life cycle level, the specific content of each document will reflect the characteristics of the product being described. Output of the detailed design phase of each level of the life cycle is to be implemented by the life cycle the next level down (except for the lowest levels that actually fabricate the products). As a result, the design specifications at one level become the requirements for the next level.

CM provides the means to coordinate the content of all technical documentation and manage change. CM facilitates change management to technical baselines and assures that organizations which are impacted by a change to an approved baseline have an opportunity to review and authorize the change prior to implementation.

### *Management and Technical Reports*

There is a requirement at each life cycle level to report management activities, development progress, and observed technical characteristics of the products being produced. There are a number of formats that can be used to report the action requests, management status, technical progress, and product deficiencies.

These management and technical reports record the development experience and document agreements and assumptions between independent organizations within a life cycle level, between levels, and between the developer and recipient of the data produced by each life cycle.

Management and technical reports may be produced by any individual or organization performing management or engineering tasks or any organization interfacing with the project organization. These reports provide the document record essential to track agreements and status. This audit trail is as essential a part of the life cycle documentation as are the technical documents. If CM provides a means to coordinate the distribution of reports, the function becomes a focal point for current information concerning status, issues and problems and achievements.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

### *Product Assurance Documentation*

The assurance documentation defines specific assurance steps at each life cycle level. These documents is to provide a consistent and continuous verification approach at each life cycle, and provide a sustained assurance structure as engineering information transitions between the individual life cycles. The specific assurance methods and techniques that are applied to the life cycles should be consistent with the planned structure defined through the planning documentation.

The assurance documentation products interact with the engineering process to provide a means to monitor the integrity of data against predetermined criteria.

Although the basic assurance documentation structure is common for each life cycle, the rigor associated with the assurance process, and as a result the documentation that goes with it, is determined by the risk avoidance requirement.

Assurance is the mutual responsibility of the sponsoring and engineering organizations. Whether it be the higher life cycle, or the organization that will apply the software to the user environment, CM provides the means to coordinate the content, release, and change of product assurance documentation.

### *The Documentation Relationships*

In any project environment, there is a two-way flow of information that affects the production of software. Approved information in the form of plans, procedures, planning requirements and constraints, and specifications and design parameters flow downward from the system or subsystem life cycle, while System Problem Reports (SPRs) or Engineering Change Requests (ECRs) identify problems to higher-level life cycles or activities which require changes to previously approved information. While specific life cycle requirements are not covered in this text, the information relationships that they support are essential components of the information framework and must be managed.

Documentation is the means by which the engineering, assurance, and management and government organizations communicate ideas, establish agreements, and monitor progress. During any phase, updates to the previous documentation may occur. These documentation segments are rooted in the information base requirement and provide the packaging and delivery format and structure that supports the life cycle requirement.

There are many ways that these segments are defined. For many projects, the fewer the number of documents used in the information base, the less the documentation cost and management problems. On the other hand, the management, production, and release of one large document that changes frequently may be a CM nightmare. The management, control, and production of many small, topic oriented, documentation segments may be preferable from a CM standpoint.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

Maintenance of the relationships; the technical integrity; and the status, release, and change to many documentation segments may prove unwieldy if it is not a focus of the project information management and control requirement.

### *The Life Cycle Relationship to Engineering*

A software life cycle treats the process of engineering as a serial occurrence of discrete engineering activities. These activities are linked through the formal specifications that describe the results of the particular phase of development. Under this approach, specifications are developed as formal documents at discrete points in the development cycle, approved by the user or buyer of the system, and from that point forward, controlled as an integrated set of project baselines. This view of engineering assumes that the process is rigorous and flows smoothly from one phase of development to the next. The problem with this is the core assumption that the engineering process allows for a complete specification of engineering that can be frozen and controlled; which is unrealistic. This is one of the reasons CM is so important in a program. It is essential for CM to provide a flexible, efficient, and controlled management, engineering, assurance, and reporting process which supports and complements the life cycle process.

Maintenance of a controlled environment between the sponsor and engineering group is the mutual responsibility of the receiving and developing organizations. To assist with this, formal reviews are held at the end of each phase in the software life cycle.

The purpose of these reviews is to allow management and other interested parties to assess the adequacy of the reviewed information to support the subsequent engineering, management, and support activities. The reviews will also assess the acceptability of those just-completed engineering products and processes.

The reviews ascertain that the technical aspects of the project are correct and understood and that the projected costs and schedule are adequate to complete the project. Formal reviews also provide the development team with a formal mechanism to obtain feedback from management and users.

At the end of each review, the documents that are presented and accepted are made a part of the formal baseline. This baseline is put under configuration control. There are two purposes for establishing a baseline:

- It becomes the formal record of the technical approach, schedule, and cost that is agreed to by management and the developer at the completion of each review.

- A baseline provides a common foundation for continued development by different organizational units or by different contractors working on the same project.

The CM process has several different segments in a program environment which manage, control change, monitor status, and provide a framework to assess the status, content, currency, and completeness of management, technical, and assurance

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

baselines. The process is required whether it controls baselined life cycle information or engineering information developed or used by the project but not incorporated in a formal program baseline.

The CM activities that control the formal documentation must establish and maintain an agreed-to management and technical relationship between the engineering organization developing or maintaining the product and the organization that sponsored the activity.

The CM procedures used at this level maintain consistency rather than control and manage engineering information integrity.

### *Government's Role in the System Life Cycle*

While the involvement of the government in the CM function will vary with the nature and the scope of the contract, in general, it performs the following functions. These activities are integral to the system life cycle and in many cases are major program milestones:

- Generates the system requirements. Program requirements are placed under CM control, analyzed by the engineering staff, and form the Functional Baseline which is reviewed and approved by the government.

- Reviews and approves the system design. The system design is presented to the government by CM as the Allocated Baseline for their review and approval.

- Reviews and approves the implementation. The completed system, including all supporting documentation, is presented by CM as the Product Baseline for test, review, and approval by the government.

- Issues or approves change requests. Change requests can be introduced into the life cycle by the government at any point. They may result from a change to a requirement, discovery of a better solution, recognition of a problem, etc. CM tracks the life of a change request from inception to closure, publishes status on the change, and ensures that the change is reflected in all appropriate products.

- Monitors the contractor's progress against schedule. CM manages the baseline schedule and all changes to it for the life of the program.

From a government perspective, CM provides the base or core set of management and technical materials associated with each life cycle phase. In addition, CM ensures that this information is current, accurate, and consistent.

### *The Contractor's Role in the System Life Cycle*

The contractor is equally dependent on the CM function during all phases of the life cycle. The contractor's CM organization ensures that the following activities are performed:

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- Requirements are baselined during the Requirements Analysis Phase to ensure that all the engineering organizations are building to the same specification. Approved changes to requirements, whenever they occur, are communicated throughout the organization and are reflected in all technical and management documents.

- The system and subsystem designs are baselined during the Design Phase and changes to them propagated through the organization.

- The integrity of software and hardware baseline versions are maintained during the implementation phase, ensuring that versions used by the engineering organization are "correct," and their contents well defined.

- All planning documents and schedules (the "how" and "when") of the life cycle are managed and controlled by CM, thereby ensuring that components are developed under compatible standards and interface at the scheduled point in time.

- Changes to all baselined items are approved, controlled, and disseminated to the community for the life of the program.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

# CHAPTER 3: CONFIGURATION IDENTIFICATION

The CM process controls information as the information moves through the program environment. As this movement takes place, it is gated by approvals which are provided through successful completion of a program review or completion of internal walkthroughs, inspections, or audits. As these events occur, the program will establish baselines to coordinate, control and manage the information which is produced or used by the program.

A baseline is a reference point from which updates and changes are applied, tracked, and recorded. Baselines are established by organizations who develop information; however, the information that is incorporated in the baseline is owned by the organization approving the information. For example, prior to a review, system engineering may release information to CM for internal project control. Change to the information may be made by the engineering staff according to established program engineering and internal CM practices, but no external approval of the change is required unless the change impacts external agreements or contract requirements. When the review takes place, and specific information is approved by the government, a formal baseline is established. Post-baseline activities require that any changes which impact approved information contained in a baseline be authorized prior to incorporating of the change into the baseline. Changes to a baseline, which do not impact approved information, do not require approval by the owner of the baseline—but the baseline is re-established when the change is made and the information content is updated.

In addition to storing, identifying, and controlling changes to official baseline documents, the CM environment must identify and track the location of all other program and project documentation. This includes those documents that are historical and will not be maintained, and documents controlled by the developer until delivery. In order to manage the data described above, CM must identify and maintain a complete inventory for each piece of engineering information or data product that comprises a particular configuration of a project. This identification process is the basis for CM controls and accounting. For simple development environments (e.g., those that are localized, non-sensitive, and static), the identification process need only control the last release of formal documentation or software. Version control of engineering data and supporting information can be informally managed. Changes to shared information may be communicated informally through informal communication, memoranda, or electronic mail. The other end of the spectrum is the CM environment that must control a distributed project, a volatile user or application, or a sensitive operational or data environment.

## *The Transition Process*

Throughout the engineering process, information builds from basic data that are produced by a method or technique, through engineering and management models that are shared within and across organizations and finally to deliverable products and

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

documentation approved by the program sponsor. Throughout this process, the data undergoes many transitions where its technical content, structure, and form are reviewed and the data component is approved for release and use. The component may just be released from one organization to another "as is," or it may be combined with other elements to form an information component. The component may also be modified to reflect an approved change.

These transitions mark a point where ownership of the data or information moves to a different organization from the one that produced or last modified it, and where the component is approved for more general use within the project. CM plays a critical role in managing these transitions. The CM disciplines provide the data repository where each individual information and data component resides in its last approved form. CM also generates and documents releases of information that have passed a transition and have been authorized for release. CM maintains and documents the status of all project information as it moves through the engineering process, and manages the relationships between discrete elements. Finally, CM provides change management over all approved components. This change-management function ensures the integrity, currency, and technical validity of all changes before the change is authorized and also after the change has been made but before it has been applied to controlled information.

There are two different project transitions that must be managed by CM:

- Baseline Transitions
- Engineering Transitions.

As described previously, the life cycle is the means by which the formal, controlled transfer of information between the organization responsible for engineering the product and the recipient of the outputs of the engineering process communicate. The process that manages life cycle information requires formal definition, management, and control of government-approved baselines.

Why is baseline CM critical to the quality of the products which are produced? The procedures a program uses to control the integrity of the technical data products and traceability to approved sponsor requirements are the means by which a quality standard is maintained. The baseline configuration practices implemented by a program is the assurance the buyer has that what has been approved is, in fact, what is built. Without this assurance, this correspondence will not be maintained.

The role of baselines is to establish a common technical relationship between the software engineering organization and sponsor or end user of the software. This commonality is established through the data products approved by the government during formal reviews that mark the end of a life cycle phase. These reviews mark the transition point in a software life cycle where the basic ownership of approved documentation and technical products moves to the sponsoring organization. Technical responsibility for maintaining the products remains with the engineering group. These reviews establish a reasonable certainty that the formally documented management and technical information is sufficient to authorize the next phase of the process.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

The formal reviews will establish specific reference points to effect traceability to previous baselines, to ensure that the product being built is, indeed, the product desired. At the review, the sponsor and engineering organization reach consensus concerning the technical integrity and acceptability of specific documentation and technical products and the engineering, management, and assurance activities used to produce them. If approval to release the information is given by the sponsor at a review, ownership of the approved information transfers from the engineering organization to the sponsor's CM organization in the form of a formal, documented product release. This release may be approved documentation, code, test information, the development records, audit trail, and CM documentation, or all of the above. From this point on, the sponsor's CM and the engineering organization's CM must retain and control identical versions of the baseline which cannot be changed unless mutually agreed to. Both CM organizations must formally control changes to approved project baselines initiated by the engineering organizations, by the government, or by one of the project organizational bodies.

The formal approval of data products at a review establishes the various project baselines or authorizes incorporating a series of changes into the approved baselines. In this context, CM provides the administrative mechanism for establishing project baselines and initiating, preparing, evaluating, and approving or disapproving all change proposals to approved data throughout the software life cycle.

Accurately identifying what has been approved by the sponsor of an engineering activity and tracing this through levels of documents and technical information is, perhaps, the most complex task facing the baseline CM function. It presupposes an ability on the part of the user to specify, in a non-ambiguous fashion, exactly what is required to satisfy the needs of the application. It also assumes that, on an end-item basis, the government and program can accurately relate and trace data and documentation from baseline to baseline. The third key assumption in implementing baseline management is that there is—in the government and program organization—sufficient expertise to adequately assess the integrity of information and proposed changes to these products.

### *Engineering Transitions*

The CM environment supports the requirement to manage and control all information developed or used concurrently by the engineering environment. CM, and its related set of elements that manage and control project information, tie the independent segments of the engineering environment together. The CM elements ensure the stability of project information within the software engineering environment. Unless this flow of information is controlled, the engineering environment concept will degenerate, and finally break down.

The engineering information that is identified, stored, and maintained by CM is that which:

- Specifies plans and procedures for system operation or system development.
- Specifies system functional and operational requirements.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- Specifies subsystem software requirements.
- Specifies the configuration of a product which has been certified to meet subsystem requirements or has been acquired from a vendor for subsequent modification.
- Describes the use of a software product.
- Comprises a software configuration that is currently operational.

There are many methods that are developed or supported by software organizations to support each of the individual areas of the development environment. Each of these has been developed with a model of information system and software development in mind which is almost certainly different from that required by the project. Each has a different set of data input requirements. Each results in an unique set of outputs and requires varying degrees of expertise and project support to implement effectively. The challenge facing the program is to select from this method "shopping cart" those that most adequately support the particular project's needs.

These individual methods must be tied together to result in a consistent technical environment. The implications associated with this definition of the technical environment are enormous. If methods do not link correctly, if they cannot be supported or if the staff cannot execute in the information system and software technical environment, then software quality and project productivity runs the risk of being destroyed.

During requirements definition, a requirements synopsis is produced for each specific set or category of requirements—user, interface, performance, operational delivery, etc. During top-level or functional design, the requirements are decomposed into a set of functional specifications which may be expressed, at the lowest level of decomposition, as a Program Design Language (PDL) description of the functional design of the software. The output of this phase also includes a description of the data requirements and the basic architecture of the database. The detail design method synthesizes a software design from the functional specifications. The output of the detailed design includes a detailed definition of the specific data requirements for the CSCI. The output from these activities is a logical design description (possibly expressed as a PDL). The coding process is performed according to the coding practices specified for the project and the output is a set of code and data. The various levels of integration and test also have specific data products which are used to transition from test phase to test phase.

At each engineering step, a particular method is applied to accomplish a specific task. Each of these steps, when completed, results in a component of one of the management or engineering models that describes a particular product or engineering step. When the step is completed to the satisfaction of the engineer responsible for the engineering step, a predetermined quality gate is scheduled through the engineering organization. If the quality gate is passed, the product(s) is released with all supporting documentation to the CM organization for inclusion into the appropriate model. At this transition, engineering gives up ownership of the data or information and cannot unilaterally change it without concurrence from the groups using it. CM builds the new release of the model incorporating the product and releases the updated model for more general project use. By tying CM to the quality evaluation and approval activities used by

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

engineering, a degree of confidence in the technical integrity of information released by CM may be expected.

### *Partitioning of CM Information*

The basic program consideration which underlies the CM requirement for baseline identification is the concept of information ownership and the requirement that the owner of a particular baseline must approve any changes to that information before the change can be made. As a result, information in a project is partitioned into logical segments with different approval required to change information included in each segment. These segments are:

- Non-Approved Program Information—All information which is under development, or being changed, that has not been approved for program use or release is in this category. The information may be changed by the individual responsible for the product without approval. The content is controlled by the individual responsible for the information.

- Controlled Program Information—All information that has been accepted for release inside the program but has not yet been accepted by the government or owner of the information at a high-level life cycle for inclusion in a baseline requires that the program establish a process for CM of evolving information which is shared within the project by different organizations. This information cannot be updated unless the change is authorized by an established project board.

- Configuration Management Information—This is the working area for CM. The area includes workspace for staging documentation and software releases; tools used by the project, project plans, procedures, and reports; and schedules, budgets, and CM records.

- Documentation Information—All versions of evolving and released documentation which has not been approved by the government as well as all program reports, action items, and discrepancy reports is contained in this partition.

- Test Information—All test documentation and versions of software released for test. The partition includes test scenarios, data, and results.

- Government Information—All information approved by the government is in this segment of the library. This area cannot be updated unless the change is authorized by the government.

These CM partitions are logical groupings of information, each having different project control and support requirements. Access to the data, control of the information, and the flow of data throughout the organization are the responsibility of CM. The products to be controlled consist of: documentation, engineering representations, source code, executable code, and status records for all major software elements. Access control and the flow of data throughout the CM organization must be predetermined and supported

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

by all management levels. The data controlled or supported by CM is either resident on automated project files or retained as hard-copy files.

### *Transfer of Data*

The engineering CM environment must provide control over all the data which meet the following criteria:

- The data product has been approved through a predetermined "quality gate".
- The data product is used by more than one organizational element in the project or in the production of more than one data product.

Program information under development or being modified is normally controlled by the individual doing the work. Completion of an internal project review, walkthrough, inspection, or audit at the end of an activity provides a quality check of the information before anyone else in the project uses it. Completion of these quality gates authorizes the release of the information from the organization which performed the work and initializes transfer of the approved information to CM. When changes and corrections are completed, the CM organization reviews the material that is proposed for release for completeness, format, and adherence to established project standards. When the data item is complete, the CM organization will copy the design material from the management, engineering, or assurance organizations to the CM files. The information will now be under configuration control and require project action via the SPR to change.

### *Configuration Control*

The approach for controlling products for use by the project is an integration of manual, procedural, organizational, and automated techniques as follows:

1. Library Transfer. System information developed by a program is of several types:

    - Management—Plans, schedules, budgets, standards and criteria which structure the program and provide the basis for conducting program activities.

    - Technical—Requirements, design, operational specifications, hardware and software products, and other technical information which describes or comprises the technical products.

    - Assurance—Quality assurance criteria, testing materials, results of quality gates, and other assurance information which results from the monitoring of product quality or process adequacy and conformance to standards.

    - Reports—Reports which document the results of a project activity describe an observed condition, document a problem, or report a finding.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- Each of the individual information elements which is developed in these categories documents a particular activity and forms the basis for other program activities. As they are completed and reviewed, CM will take possession of the products and place them into the CM segment. In this logical segment of CM, the CM organization will assemble related components of information to form a project baseline and, when the baseline is complete, place the entire baseline under control.

2. Changes to Controlled Software. The second method of transferring software from informal to controlled status is through the CR processing. CR processing consists of reviewing and evaluating all CRs, authorizing appropriate corrective actions to be taken, tracking status of all changes in progress, and closing the CRs after the changes are made. If it is determined that the recommended change affects the approved specification, cost, or schedule, the SPRs are forwarded to the appropriate project board for resolution.

- When the board tasks an organization to implement an approved change, the developer obtains the correct version of the affected software and the related documentation from the controlled CM segment. The engineering information and/or documentation is moved from the controlled segment to the working segment. The engineer will do all of his/her work on information he or she controls until the quality gate successfully completes and the product and/or documentation is ready for release to CM. When a CR is reported closed, the board will examine the corrected package for adequacy and completeness. When this package is approved, the board notifies the CM organization to accept the changed products, to complete a Version Description Document (VDD) if required, to distribute necessary materials to all affected sites, to log the CR as closed, and to incorporate the changes into the controlled segment of CM.

- When data is approved for transfer (either through board action or project reviews or walkthroughs) the CM organization will transfer the information from the working segment to the holding area of CM. While the data is resident in the holding area, the CM organization will review it for correspondence in both form and content to the requirements of the data transfer, will assemble and review all the supporting documentation required by the transfer, and process all CRs closed by the transfer. The CM organization will then generate either a new or interim system release of the required data included in the transfer and will document the release in a VDD. When processed, the data are transferred from the government to the controlled test area of CM where it will serve as the new baseline.

3. Product Acquisition. A third method for transferring information into the controlled segment of CM is the acquisition of a product for use in the system configuration. In this case, the engineering organization, after conducting tradeoff, cost, and technical analyses, will recommend acquisition of a specific capability to satisfy a specific design requirement. The developers will then review the software against the defined and approved baseline maintained by CM, and identify the specific technical and support areas affected, and any areas where the acquired product will not satisfy the needs of the project. The review will also define specific documentation,

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

development, or changed requirements that will result from use of the product; operational or interface implications; and a specific plan for interpreting or qualifying the product in light of project test plans and procedures.

- The results of these analyses can be reviewed prior to acquiring the product for use in the system configuration. CM then procures the product and the toolsmith is chartered to update any baselines, documentation, or project plans to reflect the new configuration. As with development software, the updates to the baselines will require a walkthrough to authorize software data products to move from the working areas to the controlled segment of CM.

### *Controlled Information Area*

The controlled area of CM is the project center of formal software information management as well as the repository for three categories of software information:

- Requirements and Functional Allocation—This segment of CM is the agreed-to baseline that establishes the technical relationship between the project and the government and serves as the basis for the application developers.

- Design and Implementation Information—This segment will contain controlled software design and coding data that have been released to CM through completion of specific application developers' milestones.

- Software Test and Support Facilities—Controlled software test data, tools, and supporting utility software used by more than one programmer in developing, testing, or integrating software are included in this segment.

The controlled information will shadow the organization of the working area, facilitating the smooth flow of data as it is transferred under project control. Under this structure, all information developed through, or controlled by, the project will have an identifier that uniquely identifies and documents the relationship of each data item to the overall architecture and organization of the project.

CM will integrate the data that has been moved into the controlled area into the appropriate model or baseline. This step in the identification process requires a predetermined structure for each model and baseline to facilitate the inclusion of the data into the appropriate information element. Once the data has been included in the information element, CM will generate a controlled build of documentation and/or executable products to validate the integrity of the data component in the context of the environment to which it is to be applied. In the case of executable software, a test build will be generated through CM to support regression testing. These builds will use the new components and the latest version (or selected versions) of each module required in the build from the controlled area. Modules that are not available in the controlled area will be flagged and CM will have the option of selecting from the working area replacements. If the controlled files reside in the same machine as the target, CM could then initiate compilation and linking of an executable object system, storing the system in

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

the test area. If not, a source build could be generated and transferred to the target for testing.

### *Configuration Management Area*

The configuration management area of CM is the working area accessible by the CM organization. The function of this segment is to allow storage of information management tools and data essential for CM functioning as well as storage and control of information which is undergoing the baselining process but has not yet been released for CM control. It serves as a holding area for transitioning information from the working area of CM to the controlled area. The only level of control for information contained in this CM segment is the CM organization; it may approve modification without additional review or approval.

### *Documentation Area*

The documentation area is the segment of CM used to develop and store released, but not yet approved, versions of software documentation. CM segment is an extension of the controlled area requiring the same level of information management and control. Early in the project, the specific outline for each document to be developed by the project should be defined and rigorously controlled by the CM organization in the documentation area. This will serve as basis for the developing, producing, and evaluating of all formal project documents, as well as the identification of data control and format requirements collected as the development proceeds.

### *Test Area*

The test area will contain all information released to the Test and Evaluation Coordinator. In support of this requirement there is a “shadow” CM established to control integrated software object code, documentation, support or testing information or data, and any tools or supporting information management information. CM will release three categories of information with each system release sent to test:

- Documentation that includes updates to any baseline documentation retained at the site, a VDD that identifies the content and SPR history and status of the release, and copies of any SPRs open against the releases that have not previously been sent for test.

- Software object code for each software subsystem affected by the release. There is no source code sent for testing.

- Test and support tools and data that are required to execute, evaluate, or support executing of the specific set of information to be used to quality the release.

Once the initial CM configuration is established, the CM organization may interrogate the CM-controlled information area, comparing both library configurations for exact

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

compliance. Any discrepancies can be resolved by updating the test segment from the controlled files in CM.

There are two types of information provided to CM from the testing organization and stored in the test area:

- Test records and reports resulting from the test activities for functional or operational testing, and
- CRs written by the test groups, logged by the testers, and sent to CM for processing by the board.

### *Government Area*

The government area contains all information approved by the government. The contents can only be updated through government approval of the change. Information which is included in the government area consists of documents, computer hardware, computer software, and facilities (equipment and supplies) for system operation which must be managed in order to provide a continually increasing level of capabilities to multiple users. These products form a basis for further system development or comprise a product which is either operational or is a departure point for system enhancement. As such, they are said to comprise formal baselines and are considered to be official products.

### *Configuration Identification and Formal Baselines*

If the configuration identification process is to prove adequate in the overall CM environment, there must be a consistent engineering criteria employed for technical design and development. These criteria must be used to assess the integrity of products and processes that are used to develop or support the system to be developed, before the products are even eligible for CM control. There must be defined links between independent project methods and activities to ensure a smooth transition of data as they move through the development process and the components of the CASE environment. Finally, there must be a predetermined definition of relationships between the engineering process and the development and documentation requirements of the life cycle. This definition must include an identification of tailoring requirements and guidelines to be followed when tailoring the CM environment to a software project application, and for applying a common environment for project and engineering surveillance.

The configuration identification activities have two forms: one which establishes a formal baseline for release to a sponsor, and a second which releases the models that form the engineering base to the project for use. The implementation of these steps in a project environment requires a close integration between the engineering, management, assurance, and project control project segments. There must be predetermined standards and agreed-to acceptance procedures and criteria applied to the acceptance of all products and used to establish models and formal baseline deliveries. Finally, CM must serve as the centralized release point for all baselined models and formal releases to ensure the content and structural integrity of all releases.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

CM provides two forms of support essential to the integrity of the configuration identification function. The first is to provide support to, but not control of, the engineering process. Through CM, working space is provided to the engineering organizations for storing requirements analysis data; design and code; informal builds, testing parameters and results; and parameters required to integrate and support programmer-initiated and -controlled engineering functions. These CM components are where individual data elements are developed and evaluated by the programmer before release to CM.

The second class of CM control is the formal configuration management files under project-level control that will contain the released versions of all software documentation, software, and test results not yet submitted to the government for approval. These controlled files manage released data that has been integrated into a model or a formal baseline, in controlled CM and project areas, documentation areas, test areas, and other areas that are required to assign ownership of information to an organizational segment.

While the role of baselines is common at each life cycle level, the baseline content and structure becomes more detailed, design- and implementation-based, and engineering-orientated the further down the life cycle tree the project moves. Basically, the baselines move from an operational and user focus to a development focus as the life cycles move from system, through subsystem, to software and hardware, and finally to the developer. There are several main life cycle baselines that correspond to different life cycle levels.

### *Functional Baseline*

Establishment of the functional baseline denotes the end of requirements analysis phase. All approved requirements for the software are documented in the functional baseline. This baseline may be established in two parts: one, after completion of the definition of the system operational concept describing the developers' understanding and interpretation of the government's operational requirement and environments;, and a second part which defines the developers' interpretation of the requirements to be implemented in support of the government's needs. This baseline provides the basis for definition of the functional and detailed design and allocation to the lower-level life cycle; it is not the direct requirement for the lower level baseline. This information is the basic technical information that is the basis for the software engineering effort. After the Systems Requirements Review (SRR) or Subsystems Requirements Review, the interpretation and synopsis of the requirements, the project plans, and the supporting technical information are baselined. The Software Specification Review (SSR) establishes the software functional baseline.

### *Allocated Baseline*

The allocated baseline is also made up of two segments. The first segment, approved at the Preliminary Design Review (PDR), provides functional, performance, and interface

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

requirements which are agreed to by the government organization; and are the basic definition and allocation of what will be developed. The allocation partitions the functions into subsystems if the allocation is from the system life cycle, to hardware or software if made from the subsystem life cycle, to CSCIs or HWCIs if from software or hardware. The second segment, approved at the Critical Design Review (CDR), contains the design requirements that are approved by the higher level life cycle and allocated to the lower level life cycle for further definition or implementation.

The purpose of the design segment of the allocated baseline is to provide a documented, detailed design that can serve as the basic requirements for the lower-level life cycle without further design activities. The documentation included in the baseline elaborates on the information provided in the initial functional definition provided in the functional baseline segment.

### *Product Baseline*

The product baseline is established after completion of the Acceptance Review (AR) for a life cycle. As with the system delivery, the AR consists of a Physical and Functional Configuration Audit. This review will bring the software development effort to a close. The government will review all test reports for approval and will audit all data for compliance to contractual requirements. The AR establishes agreement between the government and developer that an acceptable product, with all its required documentation and demonstrations, has been developed and is ready to turn over to the life cycle above for integration. The product baseline is the basis for controlling the content of maintenance releases from a life cycle and coordinating the requirements for system update and/or modification. The final acceptance, the point where the integrated product is turned over from the system life cycle, is the point at which the government conducts a Formal Qualification Review (FQR) to establish that the completed system is consistent with the needs and requirements of the end-user as expressed in the system specifications.

### *Additional Baselines*

While the baselines discussed above are typical of those established in a program, each program establishes a set based on project needs and requirements. The specific set is defined in the Program Standards and Conventions. It could include the following additional baseline examples:

- **Project Management Baseline:** This can contain information which describes how the development is to be managed, identifies constraints, and reports on development activities that have occurred. These data products deal with the business, administrative, and technical management aspects for the project that will develop the software system. They may take a variety of forms and formats, and may differ from software project to project based on factors such as size, complexity, schedule, reporting levels in the management structure, etc. Samples of the information which may be included in this baseline are:

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- Work Package—Documents a specific task to be allocated to an organization for implementation. The work package is the contract between the organization assigning the work and the implementing organization.
- Schedules—Includes all active schedules for the program in their current release.
- Cost Accounts and Budgets—Includes all cost accounts and budgets that are used to track expenditures and manage budgets.
- Earned Value Definition—This information summarizes the percentage assigned to each task being supported. The percentages are tied to the occurrence of specific milestone events documented in the milestone summary.
- Status Summary Form—This information summarizes reported progress against planned events. It includes not only schedule progress, but also earned value and budgetary performance.

- **Contract Baseline:** This contains all contracts, agreements, and amendments which have been made with external organizations, particularly those which contracted or control the effort being conducted. The baseline identifies what the program or program organization is obligated to deliver; the agreed-to budgets, schedules, and other constraints which must be adhered to; the amendments which have been agreed to which modify agreements; and any other contract factors which are relevant and should be monitored.

- **Planning Baseline:** This describes how the program will manage, engineer, assure, and report the status of subsystems, software, and hardwares—whether they be tied to the project or acquired from the outside. The planning baseline documents what will be done; how it will be accomplished and by whom; how the process will be managed and assured; and how progress, status, and problems are monitored and assured. The planning baseline is dynamic, changing as program conditions or requirements warrant.

### *Baseline Structure*

Each of the system and software baselines described above have three components. The first part is the documents and engineering products. These are the technical basis for the engineering process and plan, and describe the activities that take place. The second component is the data relationships that are developed and maintained as part of the baselining process. These relationships, maintained as matrices and database links, provide the traceable links through information maintained as part of a formal approved baseline. The third component is the forms and documentation that provide an audit and approval trail that manages the flow of information as it moves through the engineering process.

### *The Role of CM in Identifying and Controlling Baselines*

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

While the baselining authentication process—the review and approval of the content, structure and form of a particular baseline by a designated approval authority—is the responsibility of management and/or engineering; CM is responsible for the mechanics of actually establishing the baseline and establishing the correct ownership of the information it contains. The baselining procedure has four parts: ***release request, content definition, the build process, and documentation and release.***

- **Release Request:** As baselines are authenticated as a result of a successful life cycle review, a formal baseline release is authorized by program management. The release process may also be requested as a result of successfully demonstrating a changed software configuration through completion and approval of an engineering or acceptance test. The request is authorized by the manager responsible for the engineering process and is approved by the receiving organization. The release request identifies the documentation to be released, the specific software configuration, and the release recipients. The request is sent to CM for processing.

- **Content Definition:** The CM organization will process the release request by identifying the specific components (documentation and, if required, hardware or code) to be included in the release. The CM responsibility is to ensure the status and version of each release component. The CM organization will assure that all authorized changes and updates are applied to release components and that the versions of each related component are compatible. The CM organization will further ensure that all components are pulled from controlled CM versions rather than from working files of the engineers.

- **The Build Process:** After compiling and updating the components of the baseline build, the CM organization will combine them into a releasable configuration. If the release is for documentation, CM will oversee the production of the documents (or change pages). If the release is for code, the CM organization will assemble the updated components and combine them in accordance with a predetermined build instruction. If the release is a source release or hardware, the source is transferred to a medium consistent with the requirements of the recipient. If the release includes object, the source is compiled, link edited, and regression tested (if possible) by CM prior to release.

- **Documentation and Release:** The CM organization will document the content, status, and differences of the baseline and notify the recipients of the release of its availability. The release documentation will identify the stocklist of all engineering data included in the release by version number, open problems not corrected by this release, problems closed by the release, and differences between this release and the previous one. This VDD should be provided for documentation releases, source releases, and executable object releases. When the release is completed, all organizations using the information should be apprised of its existence and the version of the release should be incremented along with each controlled component.

By controlling the baselining process, the program will be assured that all shared information has a known content and level of acceptability. This process cannot

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

preclude informal engineering releases done within an engineering group or releases done to test a theory or concept for formal shared information, a level of consistency may be assured.

In order to keep track of the baselines and information framework, CM will maintain a separate inventory list and cross-reference that identifies the location and the information relationship for each data element that is in the controlled areas of CM. Collectively, these lists will identify the baselines, their locations, and their use as a building block for higher level controlled information elements and baselines.

The configuration identification process described above determines how rigorously the CM discipline is applied to the project environment. This process, and the change management procedures described in the next chapter, determine how much latitude the engineering organization may have when modifying previously approved and released information. There is a reasonably straightforward way to determine how to tailor this rigor. The higher the level of information that is defined to be managed by CM, the less control of the information framework is provided. For example, if the lowest level of information to be managed is the formal baselines that have been delivered to the sponsoring organization, the engineering organization has freedom to unilaterally change any underlying information. If, on the other hand, the basic control is provided to the information base, less freedom to change or modify is provided to engineering organizations; but there is greater project control over shared information or information that has been approved for use. If the control is extended down to the engineering data, the control is maximized but the CM function may collapse under the weight of information and change activity.

A second tailoring consideration that must be applied is the component that is managed as the smallest entity. The CM application may decide to only control information that has been integrated into a CSCI configuration. In this case, the controlled information base may include completed management and engineering baselines. Also controlled are baselines describing the CSCI and integrated code interim information describing CSCs or CSUs. The engineering organizations are provided freedom in developing the products, but there is an associated downside that must be considered.

The problem of parallel development discussed in part 4 of the text states that when two individuals change the same component simultaneously, integration of these independent changes when returned to CM is difficult, perhaps impossible. If the changes overlap, or are technically inconsistent, CM may not be able to resolve the discrepancies. If the smallest component that can be checked out from CM is a CI baseline, the problems associated with parallel development are guaranteed. If, on the other hand, the smallest component is the CSU, very small pieces of information may be checked out for change and the parallel development problems are minimized.

The third tailoring consideration is the effect that certain project characteristics have on the identification of information. Sensitive applications that must be secure, high risk applications and applications that are severely constrained by schedule, budget, or resource limitations must maintain an audit trail that documents the engineering process

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

for baseline certification. For these classes of engineering, rigorous control of information to a low level is required.

### *Organizational Roles and Responsibilities*

### The Government's Role in Configuration Identification

The government plays an important role in reviewing, approving, and modifying the functional, allocated, and product baselines controlled by CM.

- The first baseline to be established, the functional baseline, is a re-statement of the government's requirements as clarified and explicated by the contractor. As the allocated and product baselines evolve, the government provides review comments, directs changes, documents non-compliance, and approves/disapproves changes to these baselines.

- In addition to the "standard life cycle" baselines discussed above, the government often participates in recommending or approving changes to the planning, project management, and contract baselines.

- On many contracts, the planning and management documents (for example, the Software Development Plan and the Quality Plan) and the project schedules are contract deliverables and as such are reviewed and approved by the government.

- The contract baseline, the binding legal document(s) between the government and the contractor, is maintained by CM and modified by mutual agreement over the life of the contract.

Once baselines are developed, the government uses the information contained in these baselines to assist in the administrative and technical management of the contract. The following are some examples of government functions that depend on the information in various CM baselines:

- The planning and project management baselines enable the government to monitor and measure the contractor's performance against schedule and conformance to documented standards and procedures.

- Review of the functional baseline by the government ensures that the contractor has correctly understood the characteristics of the product to be developed. The result of this review process—approval, rejection, or modification of the functional baseline—is managed and controlled by CM.

- The allocated baseline provides the government with a specification of the system design. The design is reviewed for completeness, conformance to requirements, internal consistency and technical correctness by the government. Agreed-on review comments are incorporated in the updated allocated baseline by CM.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- The product baseline includes all elements of the delivered system. This baseline is reviewed and tested by the government to ensure that delivered products meet functional requirements. All test results and review comments become part of the product baseline when the system is delivered.

- The contract baseline provides the government and the contractor with a complete and accurate record of all changes to the original contract, ensuring that both the contractor and the government have a clear and accurate statement of their commitments.

## The Contractor's Role in Configuration Identification

In most organizations, it is the contractor's responsibility to staff the CM organization and to define the plans and procedures used by CM in creating and managing program baselines. Contractor-developed CM procedures (discussed in previous sections of this volume) define how the following topics are accomplished:

- Identification and location of the items controlled by CM
- Levels of authorization—access authority, change authority, release authority, etc.
- Baseline modification criteria—the process to issue a new baseline, or a version within a baseline
- Distribution of baseline material
- Traceability across baselines.
- Automated CM tools

The government and the contractor engage in a continuous dialog over the life of a program. This communication is predicated on the accuracy and integrity of the baselines established and maintained by CM.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

# CHAPTER 4: CHANGE CONTROL

The previous chapter described how information is entered into CM and incorporated into the information framework. Identification practices control the movement of newly developed information from the engineering organizations, to the information base, into releases, and formal baselines. This chapter describes how this information is modified in a controlled and integrated fashion. The process used to enter, evaluate and process change requests is discussed, and the way that corrections or modifications are made to controlled information, based on these changes, is described. The change management process is one segment of CM that is common for all project classes. Conversion, development, maintenance, and augmentation projects require change processing roughly patterned after the example below. R&D projects probably don't need to formally manage change unless the project warrants maintenance of traceability and baseline integrity.

## *The Change Control Process*

While configuration identification describes how the information framework is built up, the configuration control CM segment manages modifications to controlled portions of the framework based on changing technical, engineering, operational, or project requirements. Configuration control includes the systematic evaluation, coordination, approval or disapproval, implementation, and incorporation in a formal baseline, or into the information base, of approved changes. Configuration control is the integrated processes involving management, a forum for evaluating change requests before implementation, and CM in controlling the flow of information and the integrity of shared information undergoing revision.

The configuration control discipline is based on a “before-the-fact” analysis of change requirements and feasibility of alternative approaches for implementing a change; impact on interfaces, resources, and schedules; commitment to action; and assurance of controlled release for controlled information managed through CM. Before-the-fact analysis requires familiarity with system and program objectives, historical and planned configurations, and the sensitivity of software elements to change. In order to control changes, it is necessary to know what the configurations are, which of these must be changed, and how, when, where, and by whom.

Unfortunately, everyone is not always convinced of the need for change control.

“*Why is this necessary? After all, as an engineer, aren’t I most qualified to know how to fix a problem? I've got schedules and technical obligations to meet. I do not have the time to deal with this bureaucratic delay. I've got important work to do. The change is just five lines of code that I can patch. If I just change this interface slightly, the program will run twice as fast.”*

This represents the frustrations of dedicated engineers forced to slow down because of a requirement to have proposed changes approved before they may be implemented.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

But this discipline is essential if the integrity of the project environment is to be retained. The individual engineer cannot know the impact on testing, on documentation, on agreed-to user and support interfaces that may be experienced if that change is applied to an operational configuration. The programmer cannot anticipate the system impacts of just slightly changing an interface, or the operational implications of a system failure caused by a patch that is incomplete or which has not followed established quality or assurance procedures. Although appearing to be a burden, change control is the key discipline that ties the project together as information is modified to reflect project realities.

Change control is not just an administrative process supported by CM. It is a discipline that determines how the project does business and how resources are to be applied during the engineering periods of the project. Especially during test, the change control process must be consistently applied to ensure that all modifications are checked throughout the organization and with the sponsor before they are applied. Change control ensures that project procedures and independent checks are maintained even during periods of schedule and budget duress. Finally, change control assures that resources required to engineer a change are projected, reviewed, and approved before engineering begins.

The environment described in the following sections is tightly structured to control the most rigorous project environment; but the concepts may be adapted to provide a controlled environment for projects of different classes and requirements.

### *Types of Changes*

Changes may be external, submitted by the sponsor or external organizations or internal, submitted by engineering or support personnel responsible for engineering or supporting the product. These are classified as Class I, Class II, or Internal by the initial engineering review conducted by the software engineering organizations through CM.

### *Class I Changes*

Class I changes are those which involve contractual issues or formal system baselines. Class I changes are submitted by the engineering organization for sponsor approval before the change is made. These changes are necessary to maintain the integrity of system baseline relationships or are essential to operational support. These changes will impact cost or schedule if implemented. Such changes impact the following:

- The functional or allocated configuration identification (performance specifications, for example)
- The product configuration identification (design, specification, drawings, etc.)
- The technical requirements below contained in the product configuration identification, including reference data
- Performance outside stated tolerance
- Reliability, maintainability, or survivability outside stated tolerance

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- Interface characteristics altered
- Non-technical contractual provisions
- Fee
- Incentives
- Cost
- Schedules
- Guarantees or deliveries
- Other factors:
- Safety
- Operation, test, or maintenance computer programs
- Compatibility with support equipment, trainers, or training devices/equipment
- Delivered operation and maintenance manuals.

### *Class II Changes*

Class II changes are those which involve formal baselines but do not change the requirements satisfied by these baselines. In addition, when the engineering organization is working on or producing items to the same design element, an engineering change which is Class II to the originator may be Class I in its impact to the other contractor. Therefore, it is imperative that all changes be fully coordinated with the sponsor, the engineering organization, and all supporting engineering and interfacing organizations. Examples of Class II changes are:

- A change in documentation only (e.g., correction of errors, maintenance type code corrections which do not affect program logic, design or mathematical formulation, or addition of clarifying notes)
- A change in hardware which does not affect form, fit, or function
- Other changes of a minor nature.

### *Internal Changes*

Internal changes are made to the information base while under internal engineering change control. These changes do not affect delivered or formally baselined information and have not yet been released for Class I or II control. Internal changes must be reviewed and approved prior to implementation and an impact assessment made prior to assuring that formal baselines are not affected. For example, changes to the preliminary specification(s) prior to the final product baseline is treated as internal changes. Similarly, any change to products released during development testing or to a functional area during development testing are treated as internal changes unless the change affects the documentation representing the functional or allocated baselines. Changes under internal change control should be classified as major or minor. Major changes, generally, are those that affect released preliminary specification, internal interfaces, or internal schedules. Minor changes involve design errors that do not affect the approved functional or allocated design or external interfaces (minor coding errors, or editorial corrections).

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

Major internal changes should be reviewed and approved in a manner similar to Class I changes, except that formal Change Control Board (CCB) review is not required (the CCB is described in the next section). CCB review is replaced with a signature-approval list. Minor changes should be given minimal, expeditious review that does not interfere with development and testing. This might be a review and approval by the development group head. The major task of internal change control will be to ensure that the product is being built according to the preliminary specification which establishes the baseline at CDR.

### *Change Control Boards*

There may be many forums for pre-evaluation of proposed changes. The change assessment forums may be nothing more than a technically competent and responsible individual reviewing all proposed changes and allocating them for implementation. The forums may be a structure of boards that rigorously analyze and approve change requests and assign engineering responsibility. The simpler the structure, the more efficient the process, but the higher the risk that a critical impact will not be recognized before a change is made. Two change evaluation forums are the Engineering Review Board (ERB) and the CCB.

### *Control of Change to Non-Baselined Information*

The reporting of software or system requirements, design, or development problems is accomplished through a formalized procedure administered and supported through CM. The forum for this review is the ERB. This board reviews and evaluates all submitted SPRs, authorizes appropriate corrective action to be taken, tracks status of all changes in progress, and close the SPR after the changes have been made. These project-level practices control released code and documentation that are under investigation and/or being redone, but which have not been approved by the program or sponsor.

The RB is a control board comprised of technical personnel intimately involved in developing the system. Different configurations of the ERB evaluate engineering change to system, and software or hardware components. This project organization establishes and performs technical analysis on all informal baselines to ensure developmental data product integrity.

The ERB is composed of project representatives who meet to consider current system problems and project issues which are submitted on SPR forms. The ERB meets periodically to discuss each SPR which has been submitted and to call in other project members as needed to obtain information concerning a particular problem. The problem may be solved during the discussion, and thus be marked “closed” and signed off immediately. An SPR requiring additional study is assigned to the appropriate individual(s), and the action to be taken and date due are recorded on the form. Such SPRs are considered “open” until the action is taken, a written solution is delivered to the librarian and the solution is approved. The CM librarian is in charge of keeping these forms up to date, and scheduling and coordinating ERB meetings. ERB membership may include the following:

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- System, Software, or other Technical Manager, Chairman (as required)
- System, Software, or other Development Manager
- System, Software, or other Technical Task Leaders
- Test and Evaluation Representative
- Systems Engineering Representative
- Program Representative
- Engineering Representative
- Quality Assurance Representative
- CM Librarian (Recorder)
- Other representatives such as product manager and marketing representatives who have an input to product integration

The ERB is responsible for the processing and assignment of all SPRs generated against the developing software system. SPRs which are determined to be out of scope are submitted to the program CCB for formal action. The CCB may task the ERB to implement a sponsor approved change to the formal baseline.

### *Change Control to Baselined Information*

The organizational body responsible for processing proposed changes to established formal baselines is the CCB. The CCB reviews, evaluates, approves or disapproves, and releases major technical and non-technical alterations to system components, both hardware and software, and all baseline specifications. Its primary responsibilities are to conduct a complete change impact assessment and analysis, control cost and schedule impact, and assure optimum and coordinated implementation of approved changes to established formal baselines.

The CCB differs from the ERB in that it primarily deals with management issues and sponsor interfaces rather than technical issues that are the domain of the ERB. Changes that effect formal baselines are submitted to the CCB after technical review and analysis is performed by the ERB. The CCB interacts with the engineering organization through established project level configuration control practices. This joint responsibility should continue until the initiating change has been approved and released for incorporation into the formal project baseline.

The CCB is a management board that consists of project management and project technical representatives from throughout the project. CCB membership may include:

- Project Manager (Chairman)
- Software Development Manager
- System Engineer Manager
- Test Manager
- Quality Assurance Manager
- Project Control Manager
- Contract Administration Manager

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- CM Manager (Recorder).

If a discrepancy is determined by the CCB to require a contract change, it must first be negotiated with the sponsor before implementation. All of these changes will require sponsor review and acceptance, and could possibly lead to contract changes if the scope of the effort is effected.

Changes to controlled baselines are recommended via the ECR. ECRs are classified as either Class I or II and originate through ERB evaluation of SPRs or formal requests for changes to official baseline items such as documentation, software, hardware, etc. The primary attention of the CCB is directed toward the processing, planning, and submittal of Class I changes.

An ECR requests a change to a controlled software baselined document or configuration. When returned to the CCB for closure, it must be accompanied by a red-lined copy of the latest version of the document/drawing. When the board approves of the changes, the document/drawing is corrected or updated by either a change or revision. A change is shall be accomplished by issuing a Document Change Notice (DCN) and attached change pages. A revision requires a complete re-issuance of the entire document. In general, corrections to a small portion of the specification shall be accomplished by a change, whereas extensive corrections will require a revision.

ECRs that have been approved by the CCB are converted into Engineering Change Proposals (ECPs) for submittal to the sponsor. On the sponsor's side of the interface, a comparable organizational element will process the ECP and either approve its implementation or reject it. The reality of this interface is that it is normally too slow to reasonably support project requirements. As a result, the CCB will often authorize implementing an ECR prior to approving the ECP. This implementation is at risk since disapproval may mean that the engineering may have to be undone and the resources that were spent will not be recovered.

### *Configuration Control Application*

The following discussion provides an example of how change control may be applied to a project. Although not a universal application of the change control process, it is representative of the flow of paper, responsibility, and information that takes place when managing change to controlled information.

The formal approval of information at a review establishes the various project formal baselines and establishes a departure point for incorporating agreed-to changes into the formal baselines. Baseline CM provides the administrative mechanism for establishing project baselines and initiating, preparing, evaluating, and approving or disapproving all change proposals to sponsor approved data throughout the software life cycle.

Formal change control is the means by which correspondence between sponsor expectations and engineering realities is maintained. The baseline change control process has three basic parts:

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- **Initiating Change Requests**—Changes to a project's contractual item can be requested by any project personnel, by the sponsor, or by an interfacing contractor.

- **Reviewing Change Requests**—All change requests initiated within a project are initially reviewed at the lowest possible approval level within the project. If the problem described in the request is minor and does not affect other areas of baseline items, corrective action is also implemented at the lowest possible level. Problems that are more serious and have far-reaching impact require higher levels of review. The precise levels for various kinds of changes depend on the size and structure of the project.

- **Implementing Approved Changes—**If approved, a requested change is implemented. This may require changing the software product, its documentation, or both, and may also involve changes to the contract.

Changes to controlled CI, HWCI, CSCI, CSC, and CSU information may be initiated in response to a variety of events:

- **Deficiencies—**An existing baseline may be found to be inadequate or incorrect by design, implementation, assumption, or for other reasons.

- **Hardware Changes—**Problems with hardware components and interfaces among hardware subsystems may yield to solution only through software baseline modifications.

- **New Operational Requirements—**The ground rules for the system's operation may be modified (e.g., a performance mandate may be reduced or enhanced in scope).

- **Economic Savings—**Means for effecting cost savings may be determined, or lower development or operating costs may require software modifications.

- **Schedule Accommodations—**A system/subsystem implementation schedule (for hardware, software, or facilities) may be accelerated, or a pre-established implementation schedule may be found to be non-achievable.

### *System Problem Reports (SPRs)*

SPRs may be generated by project members or by the sponsor. SPR forms are kept in CM and completed forms are turned in to the CM librarian. The SPR form is used to communicate problems or requests for changes during all phases of program development. SPR processing and status reporting procedures are the direct responsibility of the CM librarian. All SPR information is directed to CM for initial action. The CM librarian ensures that all SPRs are reviewed for validity and initial impact assessment. The librarian logs all incoming SPRs, assigns each SPR an identifying number, and verifies the change classification if possible.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

The SPR form is used to communicate problems or requests for changes which do not affect controlled baselines. While the title suggests only problems should be reported, it serves an equally important purpose as the means for identifying needed changes or ways to improve the program. Use of the SPR in this way significantly reduces administrative overhead but still preserves traceability.

The CM librarian will be in charge of keeping these forms up to date and will schedule and coordinate these meetings. The librarian's duties regarding the tracking of these forms follow:

- Blank SPR forms must be available from CM in a designated location from which they are submitted by project members for consideration.
- Each submitted form must be reviewed for completeness of information and assigned a number for future tracking purposes.

To prepare for the change board meeting, the librarian:

- Pulls all newly submitted forms.
- Pulls all forms that are due for review at this meeting, along with solutions which have been delivered in response to the change requests.
- Contacts assignees who have not responded with solutions to due forms for a status report on their activities.
- Provides electronic or hard copy copies of all this material for each CCB or ERB member.

At the meeting, the librarian:

- Records results of each discussion on the form.
- Marks any ECRs or SPR determined to be closed.

After the meeting, the librarian:

- Notifies each assignee of the ECR or SPR assigned, action required, and date due.
- Collects approval signatures for all closed ECRs or SPRs.
- Pulls all forms that are due at the next meeting and sends reminders to assignees.
- Prepares minutes of the meeting and distributes to all project members.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

The forms will be important in describing current system status, and thus must be maintained in a fashion that allows easy access in a number of different ways. The information will need to be accessed in the following ways:

- All forms assigned a particular person for action
- All forms concerning a particular product
- All open forms
- All closed forms
- All forms against a particular system or software build.

Assignees for open SPRs are notified before each meeting of their obligation to deliver the solution and any other materials, such as analysis reports, to CM for closure.

### *Change Board Processing*

All new SPRs are placed on the agenda of the next meeting of the ERB. The ERB serves as the initial gate for all internally generated SPRs. The ERB verifies the classification of each SPR as a Class I, II, or internal change. SPRs for Class I or II changes are forwarded to the CCB with the ERB's recommendation for resolution. SPRs for internal changes are forwarded to the CCB for their concurrence with the disposition that the ERB has decided on. If more information is needed before the ERB can come to a decision on the classification for the change, or on what action should be taken, the SPR is referred to the ERB for analysis.

When the change board convenes, the change board decides if the ECR or SPR is out-of-scope or if it is strictly an internal change. Out of scope changes are changes that impact baselined documentation or requirements and require program level action by the ERB. The change board further decides if the SPR involves a Class I, a Class II, or an internal change; this is then indicated on the form. A Class I change involves baselined documentation and requires project CCB and sponsor approval. A Class II change is processed by the ERB, but will require concurrence by the CCB before it can be closed. Internal changes require only action by the ERB. If a Class I change is involved, the SPR is forwarded to the project CCB. Otherwise, the SPR is discussed and resolved by the software organization.

The ERB is tasked with evaluating and approving all Class II and internal changes, and supporting the analysis and resolution of Class I changes as requested by the CCB. The ERB chairman has the final authority over board decisions. He/she may solicit opinions from any board member or call for a vote if he/she so desires, but his/her decision is final. The CM librarian is responsible for enforcing the CM principles and procedures and may refuse to execute a change which does not comply with CM policy. The actions that may result from the ERB are:

- **Reject**—Reject SPR
- **Assign to CCB**—Assign to CCB for resolution
- **Assign for Study**—Assign to technical organization for analysis

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- **Assign for Correction**—Assign the SPR for correction authorizing update
- **Append to Open SPR**—Block SPR with other problem under analysis
- **Reanalysis**—Send back SPR for further analysis
- **Table**—Defer analysis but don't redefine suspense date
- **Defer**—Defer analysis and assign suspense date
- **Build System and Release**—Authorize a software build and release
- **Immediate Resolution**—Authorize release but hold SPR open until documentation and test is complete.
- **Close**—Close the SPR. Problem is resolved.

If a change is approved for implementation, CM will generate a Software Change Order (SCO) and assign it to the engineering staff to perform the change. If the change involves code changes, the amount of retesting will be determined as part of the tasking. The change board reviews the change before the update is made. The project librarian ensures that all CM library entrance criteria is met and approves the change for incorporation into CM. The ECR or SPR is then closed and copies of the completed forms are given to the originator and responsible programmer.

The CCB verifies the classification of each SPR. Class I changes, unless rejected by the CCB, are turned into ECR because a baselined configuration for is affected and sponsor approval will be needed.

After each CCB and ERB meeting, CM updates the status of SPRs and ECRs to reflect the board actions. The CM librarian notifies affected managers that the board has made an assignment for analysis or disposition of an change request. Prior to each subsequent board meeting, the librarian reminds the assignees of their obligation to deliver analysis or disposition reports to CM for presentation to the board.

A change request (SPR or ECR) is considered closed when the requested internal change has been installed and the CM library has been updated; the board has rejected the change request; or the CCB has initiated an ECR based on the SPR. When an ECR or SPR is reported closed, the change board will examine the corrected package for adequacy and completeness. When this package is approved, the change board may request the librarian to:

- Accept the changed products
- Complete a VDD (as outlined below)
- Distribute necessary materials to all affected operational environments and sites
- Log the form as closed
- Incorporate the changes into the library.

### *Interface Changes*

Any ECRs or SPRs which call for changes to approved or operational system, hardware or software interfaces must detail the reason for the proposed change, the difference and impact on both sides of the interface and all impacts associated with implementing

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

the proposed modification. If the proposed change is internal to the system and does not impact cost or schedule, it is documented on an SPR and processed through the ERB.

Irrespective of cost or schedule impact, all proposed interface changes are documented on ECRs and evaluated by the CCB. The CCB will review the proposal and convert it to an ECP if it recommends implementation. The ECP is sent to the sponsor who will evaluate it at an Interface Control Working Group (ICWG). If approved, all affected engineering organizations are notified to implement the change.

The ECR must also be accompanied by any operational documentation changes necessary. After verifying that the interface configuration is as described in the ECR, the librarian updates the interface definitions affected and notifies project members affected by the change.

### *Deviations and Waivers*

Deviations from established design, installation, and performance requirements (and from items baselined throughout this process) may be requested through the CCB for sponsor approval if alternative methods, processes, or schedule constraints preclude as-agreed delivery of the items. Products should not be delivered with any deviation from any established baseline without CCB and, ultimately, sponsor approval.

Deviations to established agreements are recorded as changes to all documentation and drawings per CM change procedures. If any part of the system does not meet contractual requirements, the sponsor has the option not to accept the system. If during the design and manufacture of the system or any of its components, minor discrepancies to the final contractual baselines are discovered and can be considered acceptable, or made acceptable after some modification, the sponsor may grant a waiver(s) which deems these discrepancies compliant with contractual requirements. At this juncture, the waivers are recorded by CM as changes and all as-built, as-delivered documentation and drawings are updated and delivered as final.

### *Change Closure*

When a change has been made, the librarian notifies the project sponsor of the new release. This notification warns of any effects the change may have on their system, and alerts them to any changes they may need to make in operating procedures or equipment usage. Internal engineering groups will be alerted of any changes they may need to make. If the change requires work on the part of the users to maintain compatibility with the system, CM will notify users in advance of the actual system release date. If the change requires developing new software modules, the development should use the requirements of the software development life cycle, and the delivery will include all design and operational documentation required for CM acceptance of newly created software.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

### *Change Control Tailoring*

The above discussion provides an overview of one possible implementation of change management. One may think, "What a lot of control; all of that is not needed." This is probably right. Small, low-risk projects may be able to integrate the CCB into a single board that is much smaller than the one described. The relationship between the sponsor and the engineering organization may be much more informal if the organizations are integrated into a single development project. The change documentation may be a single form rather than the structure described if project characteristics warrant.

On the other hand, there are project environments that may require more rigorous change management than the described process. Distributed projects, or projects that have concurrent engineering requirements, may require an integrated change control structure that is more complex than the one described. Projects that must fit an absolute operational requirement or satisfy stringent interface or performance requirements may require absolute change visibility over all baselines and engineering modifications. The point is that, although all projects require change management modeled after the example above, its implementation may be dramatically different based on project needs.

There are several characteristics that are required for most configuration control applications. First, any individual associated with the project should be able to enter a problem report to document a suggestion, concern or observed problem. And, if a problem is reported, the submitter should be guaranteed a response, even if the report is rejected.

Processing of problem reports might be centralized in a single organizational entity. The engineering organization should pre-evaluate and scope problem reports and provide the initial analysis as addenda to the problem report. The problem report should be evaluated by a single organizational element before the modification is authorized. This pre-implementation approval should be conducted through two logical forums, one that evaluates proposed changes to formal baselines and one that assesses technical impacts of proposed engineering changes.

Approved modifications might be formally tasked to the engineering organization and the implementation of the change tracked by the engineering organization. Completed modifications should be evaluated for completeness and technical integrity and approved by the same organization that approved the initial change, before a new baseline is built. Finally, CM must maintain a full audit trail; tracking the progress, activities, application, and integrity of the change process.

### *Organizational Roles and Responsibilities*

Change control is the most critical element of CM. It provides the government and the contractor with:

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- A mechanism to introduce change. Change is very often a function of time—the longer the duration the program, the more change. There must be a controlled, disciplined approach to presenting changes to the program community.

- Criteria by which to classify changes. Not all changes are equal, and there must be a method to determine importance and priority in order to allocate resources.

- Consistency across products. The impact of a change must be assessed by all disciplines within a program. All products must be appropriately modified to reflect the change.

- Timely and comprehensive visibility of all changes through the CCB or ECB.

- A historical record of all changes.

- A forum through which to adjudicate the implementation of a change.

### *The Government's Role in Change Control*

The government's role spans the entire change control process. The government may introduce changes into programs for a variety of reasons. Change often results from a review where an error is discovered that requires modification. As programs evolve, especially those whose life cycle is measured in years, users' needs evolve, and initial requirements are “overcome by events” and are modified. Tools and technologies not developed at contract award become desirable.

The government also plays a primary role in determining the resolution of changes, especially Class I changes. Where a change adversely effects cost, schedule, or requirement, the government has the authority to accept or reject the change. If the change has a major impact, very often the contractor will perform a risk/benefit study to assist the government in making this decision.

During the life of the program, the government monitors the change process to ensure that it is being performed in accordance with the documented CM plan and procedures. At major milestone deliveries, the government audits the product to ensure that all authorized changes are reflected in all applicable products.

### *The Contractor's Role in Change Control*

The contractor formulates the “rules” discussed in previous sections in this volume that govern the CM change process. These rules are documented in the CM plan and its associated procedures. Once approved by the government, these procedures are implemented and monitored by both organizations. In addition, the contractor is responsible for the day-to-day activities that are part of the change process, including:

- Logging items in/out of the CM libraries

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- Staffing and holding CM board meetings
- Publishing CM Reports
- Creating VDDs
- Maintaining the integrity and consistency of all baselines and building software releases.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

# CHAPTER 5: STATUS ACCOUNTING, REVIEWS, AND AUDITS

The Status Accounting, Review, and Audit functions of CM provide the program with active control over the content and integrity of the controlled information. These activities provide visibility, at any time, of the content and change status for all controlled items. This provides the basis for managing the integrity of all information, although the primary status accounting records are primarily useful during testing.

So far, the controlled building up of the information framework through configuration identification has been discussed. The configuration change control process also has been discussed, including how it is applied to different segments of the information framework. This segment of CM known as Configuration Status Accounting, includes the recording and reporting of baseline status and content. Status Accounting will identify all departures planned or made from the product through comparison authorized design data and the as-developed and test configuration of the product.

CM Review and Audit Support is the evaluation of information contained within CM, and project interfaces to the engineering organization, to ensure that all information included in the controlled segments of CM has been evaluated before baselining. Audits and reviews provide an accurate identification of the true state of the software and engineering products. This identification is based on actual data content and flow rather than projections or incomplete reviews.

These reviews and external monitoring activities provide the means to establish baseline visibility throughout the development process. CM reviews and audits span the entire development cycle, and validate the integrity of the project development. These reviews will ensure that data placed under control meet project requirements, standards, and conventions and that it is of sufficient quality and integrity to provide a basis for project use or use in the higher life cycle.

## *Status Accounting*

Configuration Status Accounting is the recording and reporting of product descriptions and all departures planned or made from the product through the comparison of authorized design data with the as-built configuration of the product. Configuration Status Accounting disciplines provide accurate status and schedule information through which actions affecting subsystems and functional areas are reported to appropriate project and functional managers. Documentation and status accounting information is provided by the CM librarian and sets forth the exact version of a system and component and all interim changes.

The reports and information management disciplines associated with status accounting application provide an essential adjunct to the identification and change control aspects of CM. These records provide a historical basis to allow the step-wise building of software, with each step providing a documented increase in technical quality and configuration completeness. The following discussion identifies the major reports that

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

are produced through the status accounting function and their role in managing baselines.

### *Version Description Documentation*

Each time CM prepares a build for a release of a deliverable system or component, CM prepares a VDD. The VDD serves as the specification, or inventory, of the documentation, software, and support materials making up a particular configuration. The VDD provides a stock list that identifies, by version number, every CI, HWCI, CSCI, CSC, and CSU that is contained in the release. Also included in the VDD are all SPRs, ECRs, and ECPs that are closed by this release. The VDD also identifies SPRs, ECRs and ECPs that remain open, and provides differences between this release, the last, or any selected release. The VDD may also describe the specific requirements fulfilled by the build.

CM may also release interim, unofficial, system configurations based on current official, controlled versions managed as part of the information base. CM produces VDDs for these configurations as well, and the documentation is reviewed to the same degree as for an official release.

As previously described, a significant amount of documentation identifies and describes the status and content of each baseline. This documentation identifies and specifies the plans for production, the technical content and baseline status of engineering and life cycle components included in the baseline. The forms which are maintained by CM are as much a part of the baseline as the technical specifications or Program Design Language (PDL) which describe the product. The SPRs, the ECRs, the ECPs and other CM documentation provide the audit trail essential to maintenance of the CM process. The VDD and status documentation define the content of each build provided by CM and the means to assess status.

### *Subsystem Status Summary*

The Status Summary Form documents the status of system components included in CM in the controlled and working areas. This summary identifies the development status of components and provides statistics concerning completion status of components by functional area.

### *Baseline Correspondence Report*

A Baseline Correspondence Report lists deviations from established baselines uncovered during Baseline Concurrence Reviews. Deviations in this case are recorded in SPRs. This report is provided to all managers.

### *Configuration Index*

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

These are listings of the contents of files in directories in controlled or working areas of CM. These serve as a dynamic configuration index. By monitoring these listings, managers can obtain current information on activity relating to a system component and its completion status.

### *Status Report*

Change monitoring is aided by the Change Status Report which details the status of all proposed and approved changes to system components in the Library Component Listings. The purpose of the report is to provide managers with a summary of the current status of all pending and approved SPRs. The Change Status Report is organized into two sections: Change Status Listing and Change Status Summary.

The Change Status Listing contains an entry for each successive change to be implemented to resolve an SPR, ECR, or ECP for each system component. The entry includes a brief indicator and/or comment that characterizes the status of the change.

The Change Status Summary provides the status information for each currently active SPR, ECR and ECP listed in the Change Status Listing. The Status Summary entry for a new problem report appears in the first issue of the Change Status Report after the problem report's number is assigned. The entry continues to appear in each subsequent issue of the report for at least one issue following: either disapproval of the problem report or implementation of the change, closing the problem report.

### *Maintenance of Baseline Status Information*

It is the responsibility of CM to maintain the status of all releases, engineering components, and baselines that are controlled by PSL. The status monitoring reports are a part of the release package; they are as important as the technical content or documentation provided. If the content and status of a release are not known, or if the exact content of the controlled segments of CM accurately ascertained, modification of the components will not be possible; reproducing problems experienced in the field, test configurations that must be rerun, and accurately projecting cost to complete and projecting engineering schedules will not be possible.

There is a second critical reason to maintain current baseline status. To paraphrase a saying; “If you don't know what you have, it is too easy to be optimistic as to how long it will take to finish.” The Status Accounting functions and the associated reports provide a consistent and accurate view of the true state of the project. This view is not based on someone's idea of where they are, but rather on an objective and accurate inventory of the information under CM control. Without timely and accurate Status Accounting, the identification and change management functions of CM will not be possible.

### *Reviews and Audits*

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

Reviews are held to ensure that a product is being built error free, according to program standards, and in conformance with requirements. Audits validate that this in fact has been accomplished. Reviews and audits should be conducted by CM according to the criteria described below. Internal audits should be conducted to verify that the system and engineering's actual performance complies with development specifications. Test data is reviewed to verify that the item has performed according to functional requirements. Audits should also be conducted to ensure that all deliverable configurations are consistent with contractual and other agreements made in advance of the delivery. The audits will be performed in conjunction with deliveries, authorizing the release of information.

### *CM Audits*

Two types of baseline qualification audits, and other audits conducted through CM, provide an essential validation that releases are consistent with project requirements. These audits consist of the following:

- **Functional Configuration Audit (FCA)—**The FCA measures the degree of compliance with the development specifications and test plans and procedures. Test data for the qualification hardware are reviewed during the audit. The FCA is conducted for each system deliverable and CSCI at the end of the qualification and acceptance test project and prior to their release.

- **Physical Configuration Audit (PCA)—**The PCA is a prerequisite to accepting delivery of a production system and engineering deliverable. The PCA is conducted for each CSCI prior to the start of its formal qualification testing and for each CI prior to the start of its qualification testing. An important aspect of the PCA is the audit of the "engineering release system" or the automated configuration indices and records for each baselined item. The automated configuration indices described in this plan contain the information necessary to support the PCAs for system and engineering deliverables.

- **Configuration Management Audits—**CM audits the project to verify adherence to CM procedures and standards, and to assure the integrity of the configuration management project. In addition, CM assists the project by validating the status (open/closed) of all engineering and documentation problem reports which are documented in the SPR and ECR Summary Status Reports. Through the process of validating the status reports, CM validates all changes made to the project deliverables. These audits are performed at regular intervals. The audit encompasses the configuration management function and ensures that CM is:
    - Identifying, controlling, statusing, and retrieving all engineering and documentation.
    - Providing regular backup protection and off-site backup storage.
    - Supporting the engineering environment.
    - Producing new engineering releases according to the approved procedures.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

The purpose of the audits is to audit each deliverable end-item and evaluate the results of testing activities to assure that terms and conditions of the contract have been satisfied and that all government agreements have been met. Approval of these data by the government and QA organization establishes the product baseline and essentially completes the technical development. The problems uncovered through these audits are documented and monitored through the SPR and ECR procedures. CM maintains the summary status reports and validates all closed action items.

The audits will validate accomplishing development requirements, and establish a product configuration identification through the subsystem's technical documentation. The reviews (formal and informal) are used to establish requirements, and verify compliance to requirements and a controlled design. CM requires evidence that certain reviews and audits have been passed before accepting products into CM.

The review and audit methods provide project traceability and fall into two primary categories:

- Internal and external reviews, walkthroughs, and system and engineering data while they are under development;
- External evaluations of the status and integrity of the project and the quality of the data being developed.

### *Baseline Reviews*

CM conducts several reviews to evaluate the integrity of system and engineering baselines as an adjunct to the formal baseline review process. These reviews will verify the readiness of the baselines for release. All baselines, hardware, software, and documentation deliverables will be shown to be complete, traceable and technically consistent. The desired result of these reviews is to achieve sponsor, management, and engineering concurrence that the baselines, information elements, and executable components are ready for testing and, ultimately, release. Through these reviews, liens against the deliverables may be placed that carry through use. These reviews are:

- **Baseline Concurrence Review—**At several points in the project, the baseline will be compared against the requirements of the previous one for compatibility and consistency. The purpose of this review will be to assure that requirements, design, code, and test data are consistent and satisfy development requirements stated in the government development plan or SOW. The review will start with the requirements, taking each individual requirement and tracing it through the various baselines. Any deviations will be documented on an ECR for evaluation by the CCB.

- **Technical Integrity Review—**The technical integrity of the program system will be monitored continually by the CCB and ERB. CM will prepare an ECP summary for the program manager and the government that identifies, summarizes, and categories the open ECPs and their projected closure dates, as well as documenting the corrective actions that were taken. This ECP summary will be prepared after each CCB meeting and sent to all project and functional managers.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

### *Acceptance Reviews*

As described in the preceding chapter, CM acts as the "gatekeeper" for the formal baselining of deliverable documentation and executable products by controlling the transfer of data from the working areas of CM to the formal areas. In reviewing the baseline, CM validates that the appropriate approvals and release paperwork have been obtained for each controlled item. CM will summarize statuses maintained by the library based on information collected and maintained in the organization. These statuses include but are not limited to:

- Reviewed and approved
- Reviewed with revisions pending
- Reviewed with additional walkthrough pending
- Reviewed with multiple additional walkthroughs
- Open SPRs - corrective action required
- Completion discrepancy
- Completion discrepancy needing corrective action

For each deliverable under review, whether in one of the formal reviews or in one of the intermediate walkthroughs, CM notes whether or not the material under review was approved, was established as needing revisions with no follow up walkthrough, or was established as needing revisions with another walkthrough. CM ensures that approved deliverables are placed under control in a timely manner. If approved items are not placed under control in a timely manner due to technical project or sponsor, CM notes this as an item needing corrective action.

For materials needing revisions with no follow up walkthrough, CM establishes a time frame for completing the revisions. Revisions are signaled by the sign-off of the action item list describing revisions. CM monitors the action item list on a spot-check basis to assure that the action items are resolved in a timely manner by the CCB or ERB and that the completed action item list is signed off. For materials needing another walkthrough, CM monitors the walkthrough occurrence and baselines products that have been approved.

### *Organizational Roles and Responsibilities*

The contractor and the government define the Status Accounting procedures at the beginning of the program. The visible output of the Status Accounting process, Status Accounting Reports are developed by the contractor, reviewed by the government, and used by both organizations to assist in tracking baseline changes and making technical and management decisions.

### *The Government's Role in Status Accounting*

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

In addition to internal technical reviews conducted by the contractor's engineering organization, there are three types of formal reviews in which the government organization participates: Baseline Concurrence Reviews, Technical Integrity Reviews, and Acceptance Reviews.

### *The Contractor's Role in Status Accounting*

The contractor conducts the formal reviews and audits. In addition, the contractor is responsible for the VDD, which provides the status and content of each software release. The contractor is also responsible for the other reports and summaries discussed in this chapter. The Subsystem Status Summary and the Configuration Index are used by the contractor to ensure that schedules are realistic and as an input to an earned value` progress tracking system, if required by the program. The Change Status Report and the Baseline Correspondence Report enable the government and the contractor to assess “work in queue.” A clear understanding of the nature and priority of SPRs, ECRs, and ECPs allows the management of both organizations to plan resource allocation and use.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

# CHAPTER 6: CM ORGANIZATION AND PLANNING

The planning of a configuration environment is the bottom of a complex planning tree that structures and controls the flow of information used, changed, or developed by the software project. For the purpose of this discussion, two definitions are critical: planning is "preparing for the future," while control, the implementation of the CM plan, is "making sure we have what we want when we get there." The planning process must be totally consistent at the lowest level. The CM plan must be consistent with the test plan, which must be consistent with the standards, etc.

The implementation of the CM plan must be a complete representation of the plan as adapted to the specific needs of the project to which it is applied. In this chapter, various planning requirements that shape the CM environment are discussed, and the methods for CM planning are described.

## *CM Planning Requirements and Relationships*

The planning of CM is identification and projecting the specific requirements and structure that will manage and control the flow of information. The planning parameters must be based on the following project factors:

- The type of project that must be supported.
- The physical characteristics of the project including technical product characteristics, specific project characteristics such as organization and project distribution and the requirements imposed by the specific CM to be used for the project.
- The availability of resources to support the CM requirement. This includes personnel, facilities, budget, computer facilities and external support available to the CM organization.
- The risk and potential exposure associated with the engineering of the software and delivery of the product.
- The government, system, subsystem, and engineering environment.
- The constraints that are imposed on the effort such as delivery and schedule that structure and define the project environment.

The CM planning process starts with the ultimate user of the system, who initiates the engineering process. At this level, the control of information is essential to track what has been approved for implementation and what changes have been authorized. Planning this segment of the government environment should be a precursor to the delegation of any responsibility to an engineering organization. Each organization that is associated with the engineering activities must plan and implement an information

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

management structure that interfaces directly with the higher level CM environment. The CM environment branches through each of the engineering organizations, including software. The physical and technical characteristics of the project, the type of the engineering project (conversion and rehosting, software development, software maintenance, application upgrade, or research and development) shape the requirements that must be satisfied through CM and the level of information control required.

In this context, the CM disciplines have a very real role in adding to the management and control of information. This role is to provide a project environment at all levels in the project that will:

- Improve the probability that the delivered products will meet the documented and perceived requirements, operational needs and expectations of the user of the system.

- Increase schedule and budget predictability by minimizing rework and redundant effort.

- Maintain traceability and operational integrity of engineering products as they are produced and changed, and maintain the relationship between the formal documentation and the underlying engineering information.

In order to define an integrated CM structure that will accomplish the project roles described above, at least four levels of CM planning are required. The first level, Government CM, is the responsibility of the organization that initiated and funded the engineering activity. This CM requirement is concerned with managing documentation approval, change, and acceptance rather than managing the engineering process. The Government CM environment may also require the control of information released to the operational environment. The Government CM requirement, although not in the scope of this text, is an essential starting place for CM planning.

The second level, Systems CM, manages the definition of a system design, control of lower level (subsystem, hardware, or software) life cycle information, accepted information, and control of the system integration and acceptance process. This level maintains the information relationship between the government CM organization and the systems organization. This CM discipline also manages change initiated from the government, as a result of systems engineering or management functions or from the lower level engineering life cycles. This level of CM has the coordination responsibility for all engineering CM.

A third CM level may exist for complex project environments. This level, Subsystem CM, manages implementing of subsystems allocated from the system. This level interfaces to the hardware and software CM organizations and provides the bridge between the engineering and system organizations. For this discussion, the system and subsystem levels are considered the same from a planning standpoint.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

The lowest CM level, Software or Hardware CM, controls the engineering process and the information interfaces between the different CM organizations. This level controls fabricating the products and is the direct interface to the engineering process. The software CM level is the subject of this discussion.

### *System CM Planning*

The system planning step is the critical first planning step in building or changing a system for application to a user environment. The purpose is to define a consistent and complete environment for a controlling a project. This information may not all be in a single document, or in a document at all. The important part is that the planning process takes place and that the results are written down in a place where they may be used to structure the system environment.

In order to complete these system management planning activities, the program characteristics, contract requirements, and user expectations should be analyzed and a structure which represents the worst case, or most stringent application of the requirement or project discipline, defined. This should provide the basis for the specifying the project environment and be documented in the system management planning structure, the hardware, software, and system CM plans. This phasing of resources, project disciplines, organization elements, and technical and management controls serves as the basis for allocating resources and projecting costs and schedule requirements for the program.

The system management planning activities define a common set of predetermined requirements to be applied to all subsystem and engineering levels. This planning is done early in development to laying the groundwork for all development of software subsystems and is modified on a regular basis to keep them current with changing program conditions and development realities. A key element of the system CM planning is the definition of information and approval interfaces that exist between the system CM organization and the software CM environment.

From the standpoint of CM, whether it be system or software, this period of the project is the most critical. It is during this period that the structure of the project is defined and the technical requirements and characteristics of the project are developed. During this period, the cost, schedule, and resource constraints which limit the prerogatives of the project are identified, and engineering, management, and project control procedures and techniques are defined and implemented. A great deal of information that must be controlled through CM is produced and, if the environment is not in place, the information will not be managed.

Despite the importance of this early planning to the ultimate success of the program, it is often difficult for any manager, especially the system CM manager, to find time or resources early in the development to perform these functions. The typical system activity is front loaded with myriad essential short term tasks which require the direct participation of the CM manager and his/her staff to complete. Early government reviews, system management meetings, project staffing, budget negotiations, and

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

organization development all compete for the time of the CM organization and diminish the limited resources available to the organization early in the implementation. Longer-term problems such as establishing and controlling government baselines, releasing information to the lower-level life cycles in a controlled and documented fashion, meaningful cost and schedule projections, critical information management activities, and planning and developing a controlled and integrated CM at the system level are often deferred in lieu of these short-term requirements. The hope is that, "After these crises are over, time will be available to address the pressing, long-term issues of the program." Unfortunately, the reality is that there is never time available in a program situation.

Once the manager trades short-term requirements for long-term planning requirements, he or she has created an environment for poor productivity and unacceptable quality. The commitments on the manager and staff increase, requiring more time rather than less to complete. As the project proceeds through the design and development stages, the time that the manager hoped would be available for planning diminishes. Early system management planning structures, which would have brought order to these later project phases—requirements definition, design, coding, and test—have not been done, resulting in a project environment which, at best is unproductive and, at worst, development chaos.

The CM organization must recognize the importance of this early planning to long-term project health. This planning should be equal in importance with the short-term and early technical activities and milestones. These program activities should be under the direct supervision of the CM organization. In the case of the system management planning structure, the CM organization should have primary responsibility for defining the information management and control segments of the environment and in determining the requirements for the tools, techniques, methods, and system CM procedures and controls.

This initial system planning establishes the environment and constraints which limit the prerogatives available to the software CM organization. If unrealistic in the context of the system application or environment, the software CM may find it difficult or impossible to release quality software responsive to the needs of the system environment. If the system CM environment is loosely defined, if the management, technical, and program controls are ineffective or not implemented, or if resources allocated by the program to the software project are not adequate, the quality of the software product will suffer.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

## *Software CM Planning*

The complete and adequate definition of a system environment provides a predetermined context for the software engineering environment. The system context defines what information is to transfer between each life cycle level and how this flow is to be enabled and controlled. The system, and subordinate software CM planning structure, establishes the basic environment within which the software will be developed and integrated into the system configuration and environment.

The CM planning structure should define an integrated CM environment. The software CM planning activities are hierarchical; requirements branch downward from a set of system development and control requirements. From these requirements, a software engineering structure is defined which satisfies the program goals and objectives and is consistent with system program constraints, limitations, and essential program relationships.

At the lowest level of the software engineering planning process, the requirements for software CM are documented and specific procedures for managing and controlling the process are documented. This relationship is critical to the success of the software project and the quality of the products which are produced. If software is produced in a program environment which is unclear, inconsistent, or ineffective, the quality of the software will be questionable despite the rigor of the software development and the effectiveness of the software engineering process.

The successful development of a software system requires the presence of five factors:

- A preplanned project environment tailored to the needs, characteristics, and development requirements of the application.

- A smooth blending of technology, project discipline, and development control based on firm requirements and an integrated set of project plans tailored to the characteristics of the project.

- Application of tools, techniques, and project methodologies which cover each phase of development, are linked together through project controls and data product,, and are tailored to project technical characteristics, personnel experience, and technical, administrative, and management constraints of the project.

- A smooth, controlled transition of data consistent with technical and development standards of the project; a predetermined, effective flow of responsibility as the implementation proceeds from phase to phase and as responsibility shifts from organization to organization; and a set of quality gates which monitor and evaluate the quality of the data before the impacts associated with poor quality affect the quality of the end products.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- A staff technically competent to produce the software, motivated to meet schedule and cost commitments, coordinated to ensure that the activities of one organizational element mesh properly with other segments of the project; the activities of all segments of the project are focused, directed towards a common set of goals and objectives and that the staff is committed to produce a quality product consistent with project standards, operational and performance requirements and the development requirements of the project.

Tailoring CM to reflect these requirements while adequately supporting the specific requirements of the project is the software CM planning challenge. If the system is a turnkey application, the basic requirements are defined by the user of the system. If the software is embedded in a system configuration, the specification of requirements is a result of the design specified by the higher-level life cycle. The life cycle requirement, and its related documentation requirement, is structured by the characteristics of the system which must be supported. The attributes of size, complexity, number of separate and unique components, complexity and number of interactions, interface characteristics, and functional and administrative partitioning establish the requirement for documentation and, as a result, the complexity of the software CM environment. The software CM planning structure defines the interactions, scales and adapts the structure, and coordinates, in advance, the many engineering and management activities that must be controlled through CM.

The Software Development Plan (SDP) establishes the management and project control techniques to be applied to the software project. This plan is not intended to define specifically how the project is to be developed, nor does it provide a technical discussion of the techniques to be applied. The plan provides the basic management structure and defines the basic environment to be detailed in several subsidiary planning documents.

The SDP defines what needs to be accomplished to produce each software CSCI, how the project is to be structured, and how the process is to be managed and controlled. The SDP provides for the following:

- Establishment and maintenance of baselines and milestones throughout the software development.

- Reporting procedures for status and technical control of project elements.

- Procedure for development and control of individual and integrated software components.

- Requirements traceability throughout system components and elements.

- Establishing a management and control structure for the project.

The software engineering is structured and directed through this plan. It is the purpose of this plan to establish the specific software management, control, and organizational concepts for software development. The SDP provides specific information concerning

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

software project organization, CM and control, quality assurance, and reporting procedures. The plan is the control document coordinating all elements of software management and covers the period from the specification of requirements through software development completion.

The SDP describes the management process to be employed in the software engineering process. It will describe, in a top-level manner, the CM concept, the quality assurance approach, and the testing approach. The plan will discuss the possible use of independent testing organizations, the development milestones, the role of the development organizations, resources, and structure, and the types of personnel that will be applied to the engineering effort. Finally, the SDP will identify the facilities, software engineering procedures, and controls that will be employed, as well as documentation requirements. Essentially, the SDP forces the manager to describe how he/she will manage, what resources he/she will employ, and what procedures will be used. It is a top-level document describing what will be accomplished and how the process of software engineering will be managed and controlled.

The SDP is backed up by more detailed plans, depending on the size and complexity of the development, that describe how the specific areas of the project environment will be implemented. The following second-level plans interact closely with the CM functions:

- **The Software Configuration Management Plan** (SCMP) describes how the SCM requirements and techniques described in the SDP are to be translated into tools, methodologies, a project work flow, and then integrated into the project.

- **The Software Procedures and Standards Specification** (SPSS) describes the programming practices, standards, and conventions to be followed by the project, the technical methodologies to be applied to the development, the basic internal and deliverable documentation, and internal format project reviews and audits.

- **The Software Test Plan** describes the specific procedures and controls to be used to verify the software, integrate the software into an operational configuration, and functionally qualify and demonstrate the software in a system environment.

- **The Software Quality Evaluation Plan** describes how the overall integrity of the systems development is to be monitored and reported by the QA organization.

From these plans, project procedures define how resources are to be applied to the project and controlled, and how the software project is to be measured in relation to resources expended.

### *Software Configuration Management Plan*

The Software Configuration Management Plan (SCMP) defines the CM functions to be implemented for software development. Through the SCMP, the Systems Manager will appoint a single Software Configuration Manager and direct that manager to establish formal baselines for the software development, to control changes to those baselines, to

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

report to the government and others concerned with the status of the baseline configuration, and to disseminate the specifications, engineering change proposals, actual changes, and change reports. This document will address the formal CM organization and responsibilities, rules for configuration identification, baseline definitions, software configuration control procedures, and rules for storage and release of software master tapes and documents. It will establish procedures for configuration status accounting and for ensuring that all organizations implement CM in accordance with the project standards.

The SCMP will also detail how the Program Support Library (PSL) is to be organized, administered, and integrated into the project. The file allocations and automated facilities of the PSL will be described along with the ECR and SPR processing. The SCMP is developed to define how the software project is to manage and control the flow and integrity of project data, the tools, techniques, and methodologies to be followed in implementing the CM requirements, the file structure and data configurations to be implemented in the PSL to store and control project data, and organizational requirements for the PSL and supporting boards and support organizations.

In addition, the SCMP will describe how formal CM is to be implemented within the software project organization and how it will manage the formal baselines and deliverable data products. The plan establishes the detailed process by providing a smooth work flow, a controlled flow of data, and a consistent structure for processing changes for all software CSCIs and engineering products. The plan will be the controlling document for all program and project development CM activities. It will be treated as an active, subject to change following the approval cycles and change control procedures outlined in detail in the later chapters herein. It is the responsibility of the Configuration Manager to enforce this plan and to ensure that the plan is kept current.

In order to achieve a properly integrated, staffed, and technically effective CM approach, the plan addresses the system engineering effort as a single project. It is applicable to the system and software phases of the project. Although the demands on CM will differ greatly in volume between the two phases, the essential CM concepts will remain identical. There are many outlines that are used to document a CM approach. The SCMP content describes a structure that will define a clear environment for managing the CM. The plan must also define the relationships between software CM and the system or subsystem CM environments.

### *Additional Software CM Planning Factors*

Planning the software CM structure requires defining several factors before the plans can be developed. These factors fall into four categories:

- Organizational interfaces that must be supported
- CM definition and control that must be addressed through CM
- Project characteristics
- Information access.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

These factors shape the software CM planning structure and constrain the implementation. As such, they are prerequisites to defining and implementing a software CM environment responsive to the needs of a particular project. The following discussion provides an overview of these factors and their effect on software CM planning.

### *Organizational Interfaces*

The organizational interfaces that must be supported by software CM must be identified; the information, relationships, and roles detailed; and the responsibilities and reporting relationships of each organizational component identified as they pertain to the software CM organization. What does this entail? First, the specific approval responsibilities for the systems CM organization must be identified: Who can approve a baseline in the systems organization? How will proposed changes be reported to the systems organization? How will they be processed and how will changes be applied to the approved system baselines? How are problems in the system configuration identified, reported, and processed? And, how are the lower level CM organizations notified of problems, new system releases, and documentation updates?

The system CM procedures describe these interfaces, and identify how they are to be structured in the system CM environment. These procedures become the basis for the software CM planning. The system CM "how to" becomes the "what" for software CM. A useful approach in defining these relationships is to do an organizational "point" analysis. Each organizational interface is identified before the software CM plan is developed. At each point, the data and information requirements are identified, the enabling approvals documented, and the information release and management requirements defined. This "point definition" makes the structuring of the organizational CM relationships, and the inter-organizational transfer and control of information, straightforward and consistent.

### *Configuration Management*

One of the purposes of software CM is to manage the content, the structure, the integrity, and the relationships of all information and data that are contained in the CM. The CM contains data released from the engineering organizations but not yet included in an engineering or management model controlled by the project; all models controlled by CM and released for project use; and life cycle documentation released for software project use or integrated into an approved system baseline. In order to define the software CM structure, the relationships between individual data components, information relationships, and formal documentation and baselines must be defined.

The system and software standards and conventions, the engineering and production requirements and responsibility, and the system baselining procedures must structure the software CM environment. These parameters need not be complete when the CM environment is structured, but the basic relationships must established and integrated into the structure to be reflected in the plan. The baseline relationships that tie the

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

system and software CM environments must be defined and the information content of each baseline defined completely before the software CM environment may be defined.

### *Project Characteristics*

The CM environment must be a true reflection of the actual program characteristics or it will not be accepted or effectively applied. Before the CM planning may be completed, several program factors must be defined and used to scale the CM environment. These factors include physical, programmatic, geographical, technical, and project type. The CM plan should be structured in accordance with these parameters if it is to prove effective when implemented. The basic considerations to be addressed are:

- **Physical:** The CM approach must be scaled to support the worst-case sizing and complexity anticipated for the project. The general factors that are used to scale the degree of control include the number of unique elements to be controlled, including interfaces, design objects, CSCIs and CSCs, and individual testable and compilable elements. Also to be considered in defining CM rigor are the number of internal and external software and system interfaces that must be managed, since these are a measure of complexity. The general rule to be followed is that, the larger the software, the more objects to be managed; and the more complex that software or system, the more rigorous must be the CM control. If a complete structure for CM is defined, often additional rigor may be applied by just designating a lower level of software component for software CM control (designate CSCs eligible for control rather than CSCI).

- **Programmatic:** Programmatic issues such as security, project and data sensitivity, risk, testing and release complexity, and physical project characteristics (number of organizational elements, volatility of the system project environment) have a significant bearing on the rigor applied to the software CM implementation. Generally speaking, sensitive, high-risk projects (whether the risks be implementation, resource, schedule, or technical) require far more rigorous CM than do simpler, less risk project environments. The reason is that, in a high-risk project situation, the two project factors that cannot be allowed are rework of completed information that has been released or uncertainty as to information status. Increasing the degree of CM rigor reduces the risk of both factors.

- **Geographical:** The physical distribution of personnel has a significant impact on the complexity of the CM environment. Projects that are local to a single office can require only informal communication to manage configurations. CM need not be rigorous; change management can be managed informally and releases documented informally between programmers. If the project is split across rooms, floors, buildings, or sites, the potential communication problems to be addressed by CM increase dramatically. This complexity is compounded by organizational splits between system, hardware, software and user environments all dealing with the same CM. The software planning effort must define these relationships—the physical and organizational distribution of the project—before software CM planning can be completed.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- **Technical:** The technical aspects of a software project greatly influence the planning of the software CM environment. The partitioning of design from the system to software, technical requirements that are difficult to implement and demonstrate (i.e., stringent real time requirements), difficult development concurrency, or delivery requirements are types of technical factors that can significantly impact the CM structure. The more difficult the technical environment the more rigorous the software CM must be. Complex information relationships must be maintained, traceability must be absolute, and the engineering environment is liable to be highly volatile as solutions to engineering problems are investigated.

- **Project Type:** The type of project that is to be supported by CM must be defined completely before the CM environment may be planned. Although this seems obvious, the project type definition is not that simple. Software engineering projects are not always just rehosting or conversion, development, application upgrade, maintenance, or research and development. More frequently, a specific project environment is a hybrid of several types. During the initial stages prototyping may be done requiring an R&D CM environment. Later, software engineering may take place requiring the development controls. As the product moves into testing, the maintenance CM structure controls changes to it. New builds require the functional augmentation of the product applying application upgrade control. The required software CM techniques must be integrated with the project environments and time phased so the proper control may be applied to the project when needed. These controls should be integrated into a consistent structure and result in a preplanned migration of CM control and discipline.

When planning the CM environment, it must always be structured to solve the anticipated worst-case problem. If controls are put on that do not adequately manage the CM in a software project environment, it becomes very difficult to add or augment the CM environment to adequately support the project requirement. Personnel will resist, the automated and support facilities may be inadequate to support the additional requirements, and the resources available to CM may not be adequate to take on the increased load. On the other hand, it is straightforward to reduce rigor. During planning, a useful rule is to always scale software CM to the most rigorous anticipated environment, then remove controls if necessary.

### *Information Access*

The CM for a program contains a great deal of data and information that is owned and/or accessed by different individuals or organizations within a project. The complexity associated with this ownership and access requirement is compounded by information that, in its basic form, is owned by one organization but is used as integral parts of configurations developed or supported by other organizations. Security of the CM environment requires that, within a project, specific access controls be placed on each piece of information controlled by software CM. This access should be, at a minimum, check-out, check-out for change, write access, and update access. Each project individual should have an access permission provided by CM. These access

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

permissions are for specific information included in the CM. They must be provided, whether or not the framework is accessed manually through CM or through a database management system. The basic structure of permissions and the way of granting them through CM must be defined as part of the CM planning environment.

### *Final Considerations*

The number of planning documents called for in this section is significant. Although the documents are the preferred presentation media, the purpose of the plans is to define a working project environment, not write a document. Changes in planning scope and project structure must be made for smaller efforts. Planning documentation selection and content is based on the engineering process to be used in developing the system and software components. This process is defined and adapted based on the requirements of the application, the size, support, and technical characteristics of the product and the overall project support and reliability requirement of the application.

The role of the planning documentation and the underlying data is the basic criteria to be applied when defining how the information used or developed by the project is to be controlled and how these controls are to be applied. For every major software element, there should be at least a Software Development Plan. In that plan, the scope of the documentation effort will be defined. There should be only one SCMP and QA plan for the software, not one for each individual component. If additional CM planning, for example, is required, it should be included as an appendix to the basic plan. Further, including other pieces of information by reference, although useful in augmenting the content of a document, requires that the impact of changing a document be assessed on all those that are referenced. In no event should information be duplicated in more than one document, or maintenance of the integrity of document content will become unmanageable when a piece of data changes.

### *Government's Role in Planning*

Having the government take an active role in planning the CM process with the contractor is critical to the success of the program. The government and the contractor work as a team to plan the CM process. As part of this process, they consider the unique physical, geographic, programmatic, and technical aspects of the program.

Initially, the government defines the CM requirements. These may be simple or complex depending on the nature of the program. On many programs, the government specifies the form, content, scope, and schedule for all deliverables. These deliverables affect the entire program life cycle and include: the planning documents, design and code review material, acceptance test reports, and audit reports. The government reviews and approves/rejects the major planning documents at initial submission and at subsequent version releases throughout the program life cycle. In most cases, the government and the contractor discuss modifications to planning documents prior to release, and rejection is unusual.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

Finally, as part of their input to the contractor's CM planning process, the government ensures that there is a vehicle through which compliance with documented plans and procedures is monitored. There is a complementary CM function within the government's organizations which requires similar planning, documentation, review, and audit. The government must be prepared to receive and maintain the products built and delivered by the contractor.

### *Contractor's Role in Planning*

The contractor, with input from the government, has the primary responsibility to plan the CM process. As part of the CM planning activities, the contractor:

- Writes the major program plans and subplans that address CM processes (e.g., the Software Development Plan (SDP), the SCMP, the SPSS, the Software Test Plan, and the Software Quality Evaluation Plan.
- Ensures that the contents of the plans and subplans meet government requirements and expectations, as well as budget constraints.
- Ensures that there is a direct and seamless interface between the CM organization, the technical organization, and the government.
- Evaluates and procures CM tools and technologies suited to the program.
- Defines a mechanism for assuring that the CM process is implemented as planned.
- Defines a process to ensure that the planning documents are “living documents” which are maintained over the life of the program.

In addition to the administrative aspects of CM listed above, the contractor is responsible for ensuring that the defined CM process maintains the technical integrity and consistency of all configuration managed program products.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

# CHAPTER 7: SAMPLE CM APPLICATION CONSIDERATIONS

The CM environment brings significant advantages to a project environment. There are many problems and application challenges that must be addressed if the disciplines are to provide management of the controlled segments of the information framework. The most significant of these is the consistent management of information in an environment that changes and adjusts in a seemingly uncontrolled and dynamic fashion. The control of information being developed and changed by many organizations, each with different motivations and agendas and all sharing severe schedule and budget constraints, establishes a project environment not quite as perfect as the one described in the previous chapters.

This chapter will discuss several software CM environments and problems that must be dealt with when applying the discipline to a real project environment. The problems discussed in this chapter include:

- The data explosion and dealing with engineering resistance during cost and schedule pressure
- Dealing with a distributed development environment
- Controlling large amounts of information while still being responsive to project CM requirements
- Controlling object-oriented projects using Ada and the problems of parallel development.

## *The Data Explosion*

Programs usually begin with a small staff of engineers who are knowledgeable about the system environment that the program must support. The team is small, the information that they produce is limited and reasonably static, and the number of interactions with project organizations that are actively using the information being produced are limited. The interaction with the sponsor is frequent, CM plans have not been completed or implemented and the PSL and CASE environment has not yet been fully implemented, or applied. The problems that are associated with the seemingly undisciplined CM environment are compounded by the whirlwind of activity that is facing the CM manager and his or her staff. Schedules must be generated, budgets must be negotiated, staff must be hired, and daily requirements of supporting other organizations leave little time to plan an structure a long-term CM solution. The software CM organization only has three organizational relationships to support:

- The software organization
- The sponsor
- The systems management group.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

The typical CM manager will use the easiest solution to structuring a CM environment to solve his or her immediate requirement. The planning and PSL implementation will be deferred until time will permit its completion. A stripped-down CM environment that is adequate to maintain the sponsor relationship but not adequate to support an active engineering effort will be implemented. Because of the limited project activity, and the restricted and static information framework the structure may prove adequate and structuring a more robust CM may be further deferred.

As engineering proceeds, the number of organizational interfaces will expand dramatically. With the increasing organizational complexity, coupled with the more complex information framework and the increasing concurrent use of information, the information will quickly overrun the ability of the early CM structure. There is much activity completing the CM plans, implementing the PSL and enforcing the defined identification, change management, status accounting, and applying reviews and audits for evaluating the baselines and controlled information.

Unfortunately, unless the manager or librarian who is responsible for structuring the environment has experience, the CM adaptation may be incorrect for the type of project or the application of the CM methods, CASE environment, or PSL procedures will be scaled to the current project requirement, not projecting the future anticipated engineering and information support requirement. As with the earlier implementation, the trap is that the CM environment will prove adequate and no further planning or structuring will take place. In this circumstance, the scenario described in the following paragraph is probable.

When it enters the CSU design and implementation periods, the project will experience an information explosion. Many parts of programs are being worked on concurrently by different engineers. These programs, and the requirements and design that preceded them, are all changing and the relationships between these changes must me maintained if the integrity of the software is to be maintained. The work of one engineer affects the integrity of many other engineers and the CM environment, as implemented through the PSL, must be able to support these relationships. The complexity of the engineering environment has grown to significant proportions and, unless planned for, the PSL will not be able to adequately support the project needs. In short, the amount of information that must be managed by the project has outstripped the ability of the project to control it.

The inability of a project to keep up with the amount of information that is controlled by CM, and the change traffic to this information, is a common and catastrophic problem that may destroy the credibility and effectiveness of the CM environment. Projects that perceive themselves as severely constrained (either by cost, schedule, or resources) will be sensitized to any delays associated with support or administrative organizations. Unfortunately, CM is often placed in this category, treated as something to be tolerated rather than an essential component of the engineering environment.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

The result of this may be a lack of truly integrated CM in the engineering environment. Control of the information base may not be through the PSL but may remain with the engineering staff. As the project proceeds and the control problems associated with the CM of baselined and controlled information increase, the role of CM may diminish.

### *Analysis of "Data Explosion" Problem*

The data explosion problem is a common, often unavoidable one that must be anticipated when the environment is planned and initially implemented. The scenario described above represents an evolutionary application of the CM disciplines characterized by constant surprises, frequent lapses in support, and a less than effective interaction with the engineering organizations.
Three factors lead to this situation:

- First, management is not committed to the concept of CM and the need to centralize support in a PSL. The lack of a commitment to planning, the inability of the CM organization to project, acquire, and commit adequate resources to manage information are clear indications that management tolerates (rather than fully supports) the centralized control of information. The lack of "buy in" by the engineering organization concerning the CM disciplines and the benefits to be accrued by rigorously managing shared and approved information point to a management attitude that will emphasize the engineering process rather than the production and release of quality products consistent with the needs of the sponsor of the activity.

- The second factor is the lack of planning emphasis that is exhibited by the described environment. Software engineering is the complex interaction of many disciplines-- management, technical, administrative, and assurance-- that results in the predictable engineering of a predetermined product. The core of this structure is the controlled production and management of information leading to, and resulting in the configuration of the end product(s). This environment must be preplanned, as must the interactions and project structure that links the project together. In the previous scenario, the lack of a commitment to planning this environment from the top is evident. The CM planning problem is a single manifestation of a much broader problem: a commitment to structure a project environment that projects, rather than responds to project situations. In this environment, the data explosion characteristic of software projects will almost always catch the CM environment unprepared and result in a project crisis centering around the ability of CM to support the information management requirement.

- The third factor is the experience level of the personnel who are managing and supporting the PSL. This problem is common because of the perceived role of CM as a clerical—rather than control—discipline. In order to understand why things are done a certain way when applying CM, how to apply the disciplines to specific project environments and what to do when problems emerge; direct project experience is essential. It is difficult to represent the dynamic nature of CM in a classroom environment and leave the student with sufficient practical understanding

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

to deal with changing project requirements. It is also difficult to convince management of the importance of having technically qualified personnel to support the CM environment rather than treating the position as strictly clerical.

In order to be effective, the PSL librarian and staff must have broad familiarity with the software engineering process. Although they need not be programmers, they must understand the engineering process; the flow of work and responsibility that exists in the project and the content, structure, and technical relationships that are part of the information framework. Besides being a good administrator and able to interact with the CASE environment, the librarian and staff must have sufficient technical understanding to avoid being rolled over by the engineering organizations. Finally, the librarian and staff must understand how to plan, how to organize the library and implement the CM environment, and how to retain the CM discipline even when project realities make this difficult. The quality of the CM staff has a direct bearing on the effectiveness of the CM environment. The following situation is all too typical of the CM personnel problem: inexperienced personnel are being asked to solve the most complex and difficult task in the software engineering environment; managing and controlling the information framework as the framework evolves and is modified. The problems experienced are representative of those that will result if the personnel shortfall is significant.

### *Potential Solution to the "Data Explosion" Problem*

The problems experienced in the scenario are typical of those that may be experienced when the three project factors interact with the CM environment to inhibit CM effectiveness. Solving the problems in a project environment is the wrong answer. By the time the solution is evident, the integrity of the information framework will be violated and the integrity of the CM environment will be compromised. This class of problem, dealing with the information explosion, must be anticipated and the project CM environment must be scaled to it. First, the planner must ensure that sufficient time, resources, and commitment exist to plan the software CM early enough to control the initial entries in the information framework. The class of software and application characteristics must be defined and understood. There must be a documented software project environment in place to provide a context for CM planning. If the CM manager does not have time or sufficient understanding to complete the CM planning, he/she must be able to delegate to competent personnel, to allow planning to take place.

There must be a defined work, information, and technical flow to allow scaling of the worst case project CM environment—probably the coding and integration periods of the engineering process. This scaling is applied to the CM planning effort and used to define the rigor, resources, and support requirements that will be used to plan the CM application. There must be a consistent and enforced commitment to managing all levels of information. This commitment must start with the manager and branch downward through the organization. Finally, there must be predetermined contingency plans that deal with unexpected or anomalous project situations. The application of CM must result in a way of doing business or the environment will not work.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

### *The Distributed CM Environment*

Much of the engineering process is modeled after a single organization building an integrated product that fulfills a well-defined operational need. Although this model is common, it is by no means a universal environment that must be addressed by the CM disciplines. A more general view of software engineering is the concurrent development of several CIs, by different organizational elements, often geographically distributed, to satisfy a perceived, rather than documented, need.

If this environment is not adequately supported through a tailored, responsive and integrated CM structure, the predictable and productive engineering and delivery of software may not be possible. When engineering is distributed, information, by necessity, must be shared. Site 1 may require a version of software or documentation that is the same or slightly different than that required by site 2. Problems may be simultaneously observed at multiple sites and corrected by each without coordination or control. Site 3 may modify software or documentation to correct a problem without consulting site 2, even though the information is shared.

Baselines may be released to the sponsor that do not reflect the current configurations available from each site or, worse still, include components that support one site but not the others. Finally, in a distributed environment, there is a real danger that two separate organizations may simultaneously update the same component of a configuration without coordinating the changes. This problem, addressed through the parallel development structure described below, is especially acute when dealing with distributed CM.

### *Analysis of the Distributed CM Problem*

In the above scenario, the CM disciplines must not only control information in a software project but also coordinate the information frameworks of multiple project components. Distributed CM must integrate the components from different organizations into an operational configuration for delivery to the sponsor.

This environment places a much more difficult requirement on CM; the need to coordinate distributed CM activities, rather than just control the information framework, in a highly dynamic and unstructured sponsor environment. In this environment, CM is a hierarchical structure with separate CM organizations supporting different segments of the project. Each segment may have a PSL that controls the local information framework that is coordinated through a centralized library that interfaces to the sponsoring organization.

The CM environment described in the scenario attempts to decentralize control while still managing the integrity of shared information. Each organization has responsibility for their own information. Coordination, integration, and release is tomorrow's problem, dealt with on an as-required basis. Although CM may be extremely effective within the organizations themselves, in aggregate, the process is inefficient and prone to fail. If the organizations are separate, but geographically localized, information control may be

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

possible through frequent status and coordination meetings held between the organizations. These meetings may be through a centralized board similar to the CCB or ERB. If geographically distributed, the coordination must rely on communications channels and agreements that are central to the location or site. When decisions or engineering modifications must be made concerning information controlled through CM, the decision will reflect the requirements of the location, not necessarily the broader requirements of the program. The effects of this narrow perspective will not be evident until the individual information components are integrated; too late to avoid productivity and integrity impacts.

### *Potential Solutions*

The CM environment provides a coordination point to centrally control all information that is released, problems that are experienced, and corrections that are assigned. The solution to the problems presented in the scenario, although conceptually straightforward, are difficult to implement in a project environment. The software management organization must retain primary responsibility for all CM and information supported by the project. Their CCB acts as the only change management interface between the sponsor and the engineering organization. Their ERB provides the facility to analyze all technical problems received on SPRs and allocate them to the various engineering locations for correction. Their PSL releases software configurations to the engineering locations for correction. These releases include object code for the basic configuration and source and documentation for the code to be modified. The releases must follow the release management procedures. Finally, the central PSL must build and document all releases—incorporating, compiling, and configuring all software received from the engineering locations and released from the project.

Each engineering location should have a local ERB that meets regularly to resolve technical issues to all software and documentation. If the change is to locally engineered products, the SPR will be processed through the engineering organization and sent to the central PSL for recording and inclusion in a project release. If the change spans or exceeds the local responsibility, it is sent to the central PSL for evaluation and processing. Likewise, the central PSL will send SPRs to the engineering location if it affects software or documentation under its control.

All releases from the engineering location should be audited, documented in accordance with status accounting procedures, and sent in the form of source and documentation for release elsewhere in the project and, if required, to the sponsor. By integrating the CASE environments that are at each location together under control of a file server, information transfer may be fully automated and the procedures of the project may be enforced through the automated CM facilities of the CASE environment.

### *CM Control of Large Amounts of Information*

Most software engineering deals with the management of substantial, but reasonably sized information frame works. In aggregate, the volume of information represented by

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

the engineering data, the information base, the documentation, and the source, object, and support facilities is significant; but certainly within current technology to control. Using a CASE environment, access, update, and on-line version retention of releases is not only possible but reasonably commonplace.

There are, however, certain project classes that cannot easily be supported within the current technology. Software maintenance, conversion, or augmentation projects that must support a large and diverse user community may require the management of, and access to, inordinately large information frameworks, all which must be accessible and controlled. Development projects that are supported by large reuse libraries of engineering or interface requirements may also require on-line access to large amounts of information.

Problems associated with managing and controlling large amounts of information include storage, CM response, and maintenance of information relationships. Unless planned for, the storage of the large amounts of information that may be required for projects as described above may be a nightmare. The CASE environment may continually run out of accessible storage, requiring frequent unplanned archiving, purging, augmenting memory, and disk or media clean up. Critical records or historical information may be removed, memory may become harder and harder to install and eventually the project CM may collapse from the information it must retain.

As more information is stored in the CASE environment in an unplanned manner, there is a point, beyond which, accessing and updating the information and the underlying data will slow to an unacceptable point. In the project CM environment, inefficient access or unacceptable response to critical information not only will slow the CM processing but will also have a major effect on project productivity, as engineers wait for information that should be quickly available through CM.

Finally, in a CM environment, critical relationships must be maintained between independent segments of the information framework. As the framework becomes larger and more complex, the maintenance of these relationships will become more difficult, reaching a point where they may not be supportable. Traceability and test relationships, relationships between engineering data, the management and engineering models and the formal baselined documentation, code, performance, and functional budgets and allocation may break down and be difficult to reestablish in an unplanned, large information management environment.

### *Analysis of the Size Problem*

In many project CM applications, external CASE memory environment is a critically limited resource to be budgeted and managed just as CPU cycles, RAM, and other project constraints. Information storage, partitioning, and access strategies that are scaled to the worst case information and access load must provide a consistent and integrated structure for managing information under CM to be applied throughout the engineering cycle.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

Unfortunately, current information management technology may not be adequate to support certain large-scale information frameworks in a fully implemented CM environment. Access demands may be too rigorous, storage requirements too extensive, and the need to relate information in certain ways may be incompatible with the size of the information framework to be managed through standard CM approaches. In these cases, different and innovative methods to structure, access, and manage information are warranted and required. They normally cannot, however, be tacked on to the CM environment; their implementation and application must be preplanned. Long lead times for acquisition of hardware, long development times for special-purpose software, and the need to develop and implement procedures compatible with the new approaches all force the need to anticipate, rather than react to, the need to manage excessively large information frameworks.

### *Potential Solutions*

The information warehouse uses banks of high-capacity disk storage to store all software, documentation, transaction records, and engineering audit trails on an accessible media. Software takes information from external sites and stores it in a predetermined place in the storage facility. The warehousing environment is a front-end configuration management facility that will take information from the warehouse, relate it together in different releases which represent views of the information, and return it to the warehouse in an accessible fashion and form. Integrated into this front end is problem reporting capabilities, extensive release management capabilities, status accounting and report generation support, and templates and support for conducting and evaluating baseline evaluations. When used in this environment, the front end also becomes the staging and release area for information to be sent to users of the facility.

Information is downloaded from the warehouse facility. Once in the front end processor, site configurations are generated, the documentation is assembled in electronic form and the entire package is downloaded to the site. Sites are then required to maintain their own CM support; returning information to the warehouse facility only when it is in a releasable configuration.

In other CM environments that face extensive storage requirements, the front end processor could serve as the core of the CASE environment supporting the CM requirements. The warehouse and its associated software would the serve as a repository and staging area for information not being currently accessed. The active files would be retained in the front end. Information relationships, release management, and problem reporting would be supported by the front-end processor. In this fashion, access times would be reduced by minimizing the information domain of the CM environment while still providing on-line storage for the entire information framework. Using the warehouse software, look-ahead searches for information resident in the warehouse will minimize access and retrieval time.

### *Control of Object-Oriented and Ada Projects*

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

CM of software that is developed using Object-Oriented Design (OOD) and the Ada language is different than that applied to more classical engineering methods. This is a reflection of the engineering characteristics inherent in the methods: modularity, interface definition and management, and information hiding. When applying OOD and Ada, the individual CSCIs are usually large, and divided into many independently compiled modules. Although not altogether different than the CSCI, CSC, and CSU model presented in the text, the strong data typing and sharing of subroutines across module boundaries according to hierarchical interfaces makes the maintenance of information relationships difficult in a dynamic project application of CM.

The modules, or packages in the Ada terminology, may each be separately coded and compiled independently. Each package employs information hiding and has a direct relationship to the design object that defined it. This relationship must be actively maintained through CM even as design, code, or basic requirements are changing. In Ada, the package may be broken into two parts—the package specification, which represents the interfaces to the package, and the package body, which implements the functionality. The specification and body may be two separate files managed through CM.

What effect do these language characteristics have on the software CM problem? First, the splitting of interface definition and implementation detail provides a “black box” view of the module to the outside world that is managed by CM. By controlling and managing software configuration integrity at the interface level, the PSL librarian is, to a large measure, freed from having to rely on code reading to evaluate the integrity of integrated configurations or software builds. Once the interfaces are straight, the engineering organization and the interface checking segments of the compiler will enforce the technical relationship to the package body.

In its complete form, Ada is far more than just a language; it is a complete engineering environment that includes a file structure and access rules, embedded file management and attributes, and integrated archiving facilities. The Kernel Ada Support Environment (KAPSE) and Minimal Ada Programming Support Environment (MAPSE) implied a new, and in some cases revolutionary approach at software engineering. More classical approaches: top-down design and programming, the standard life cycle and engineering flows, documentation standards, and the standard product assurance and testing methods and techniques were not directly applicable to the Ada environment.

This rethinking of the engineering process, while probably overdue, has a significant impact on the integrity of the engineering process, and the effectiveness of the CM components of this process when applied to a project environment. As has been seen in previous chapters, all components of the software environment must work together if the environment is to be complete, integrated, and effective. Technology must be scaled to the project to which it is to be applied and its definition must include not only a definition of individual methods, but the integration of the methods into the overall environment, rules for applying it, procedures for using it and tools to support it. Management methods should be based around the information that is produced and in the control requirements required to assure the integrity of the information and the process used to

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

engineer it. The assurance components of the environment provide the checks and balances that assure that what is being produced is what is intended and will prove acceptable. Finally, the reports provide the audit trail and visibility into the effectiveness and status of the process used to engineer the software. The single project discipline that ties these independent components together is CM.

By changing the technology and support aspects of the environment, without taking a complete look at all aspects of the engineering process and the information and control interfaces relationships between them, the risk is that, at the most basic level—the management of the information framework and flow—the discipline will prove ineffective.

### *Analysis of the Ada CM Problem*

Applying CM to projects using OOD and Ada requires a complete analysis of the engineering environment and the relationships between independent elements. The flow of work and information, the links and relationships between elements in the information framework, and the control relationships that must be exerted by CM to control and manage the engineering process must be defined, understood, and used as a precursor to all planning, especially CM. Standards for information must be defined and linked, and specific criteria for engineering data, information contained in engineering models, and baseline and documentation acceptance must be defined so they may be applied to quality gates and reviews that enable the CM flow.

Finally, a CM environment that is comprised of tools, rules, and procedures specifically adapted to the project class, characteristics, and physical environment must be planned and implemented. The planning requirement is a redefinition of the software engineering process as required by Ada and OOD methods. Partial definition may obviate effective CM.

### *Parallel Development*

In a project environment, many software engineers and support personnel reporting to different organizations are working on the same information framework. Different engineers may be working on the same CSCI, CSC, or CSU concurrently, either to build up the product, or change it based on project requirements or as a result of operational discrepancies. Ideally, this processing is serial, with each piece of information being worked on individually and not ever being concurrently changed. Unfortunately, there are projects and situations where this is not possible. To deal with schedule or budget pressures; to analyze, evaluate, or correct problems; or to fully use personnel, it is often necessary to allow two engineers to simultaneously work on the same component of documentation or software, relying on CM to integrate the two components into a single entity. This integration may be a nightmare. The changes may overlap, they may make different design and implementation assumptions, or they may be incompatible.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

### *Parallel Development Analysis*

When discussing this problem, it is important to first explore what will not work: parallel development problems will not be solved through the application of automated tools. Although certain tools, like compare facilities, will provide differences between two or more versions and version control software will identify the specific release used by each engineer; the control of simultaneous change is a difficult, and often manual process.

In the scenario above, the easiest way to manage concurrent change is to preclude it altogether. When assigning work, the engineering components should be broken into small enough pieces to allow a single programmer to work on just one piece. If the smallest component is a CSCI, test case, or document; the parallel development problem is significant and continuing. If, on the other hand, the smallest element is a CSC or CSU, the parallel development problems are minimized, but the complexity of the CM environment increases proportionately.

The place where automated CM tools can best be applied to the minimization of the parallel development problem is in managing the additional complexity associated with lowering the level of information control in a project environment. Additionally, the CM segment of the CASE environment may control concurrent access to information if that access results in change. Components may be checked out for change by a user of the CM environment. When a component is checked out, the component is "locked" and cannot be checked out until the locked status is released by the CM software.

### *The Government's Role in Preventing CM Problems*

The government can do a great deal to mitigate the potential CM problems. The Request For Proposal (RFP) published by the government contains not only the technical specification for the program, but also other requirements (e.g., program milestones, deliverable documents, personnel requirements). These “other” requirements often drive implementation decisions and can increase or decrease the risk associated with CM.

- For example, if the schedule demands an SRR on a medium to large program three months after program award, there is insufficient time to staff the CM organization, write the CM procedures, establish the CM environment, and put the requirements under CM control. Likewise, the government can have a positive influence on the choice of CM personnel by clearly defining the CM position qualifications in the RFP. The government must recognize the impact of all program requirements on the CM function during the program acquisition phase.

- The government must be committed to the principles of CM. If the government indicates its support of CM through the RFP and its meetings with the contractor, the contractor will respond. Adequate financial resources allocated for the CM function is a clear demonstration of the government's support.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.

- The government should require not only initial CM plans and procedures, but a periodic review of these documents to ensure they meet current program needs. Review of this material means allocation of resources, but prevents divergence of the implemented process from that in the plan.

The government and contractor often avoid realistic risk identification. Risk analysis and mitigation, which should be an integral part of the program management process, should include periodic review of the CM function. All large programs contain some element of CM risk. If these risks are identified early, their mitigation will be more effective, less costly, and less disruptive. Failing to deal with an element of risk does not make it disappear.

### *The Contractor's Role in Preventing CM Problems*

The contractor can take similar actions to mitigate potential CM problems. The proposal process itself causes CM problems. Contractors attempt to present the best solution at the lowest cost. Proposal managers need to understand the critical nature of CM and review the proposal for its impact on CM. For example, if the best programmatic solution is a parallel development implementation, the scope of work for CM increases.

The contractor needs to develop program-specific CM plans and procedures. Mechanisms must be defined to periodically update and distribute this information to keep it current with changing program requirements. An active risk identification, analysis, and mitigation process is a necessary component of a successful program. As CM issues arise, they should receive the same management attention within the contractor's organization as technical, schedule, and budget risks.

Staffing the CM function should be subject to the same rigor as other technical disciplines. Building and maintaining complex CM environments requires professionals with experience in CM techniques, tools, and technologies. The first responsibility of the CM staff is to acquire appropriate tools and train the development team in their use.

From the contractor's perspective, CM is a "pay now or pay later" situation. If the management team neglects CM, ignores planning, and does not have skilled personnel, delivery of a quality product on schedule is virtually impossible.

Copyright © 2002 American Systems Corporation. All Rights Reserved.
For More Information: www.2asc.com, www.iceincusa.com, www.spmn.com



WARNING – Proprietary/Confidential Information Belonging to American Systems Corporation
Use or disclosure of this information is subject to the restriction on the title page of this document.