

AD-A083 121

DOUGLAS AIRCRAFT CO LONG BEACH CA GOVERNMENT AVIONIC--ETC F/8 9/2
CONFIGURATION MANAGEMENT PLAN FOR IDAMST SOFTWARE. ADDENDUM 5.(U)

JUL 76

F33615-76-C-1297

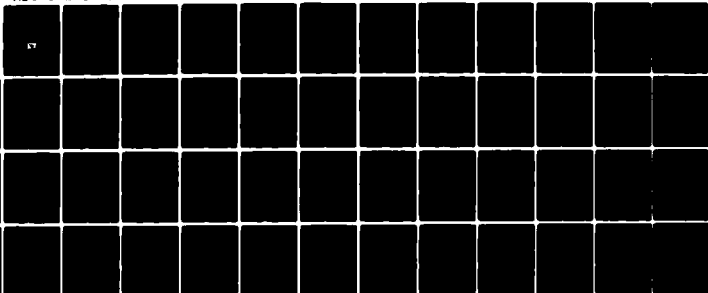
UNCLASSIFIED

MDC-J7278-ADD-5

AFAL-TR-76-209-ADD-5

NL

[OF]
AD-A
083121



END
DATE
FILMED
DTIC

209

LEVEL III

ADDENDUM #5

**CONFIGURATION MANAGEMENT PLAN
FOR
IDAMST SOFTWARE.**

**GOVERNMENT AVIONICS DEPARTMENT
DOUGLAS AIRCRAFT COMPANY
3855 LAKEWOOD BOULEVARD
LONG BEACH, CA 90846**

30 JUL 1976

CDRL 5 DATA ITEM D1-E-3108/C-118-1

CONTRACT F33615-76-C-1297

ADA 083121

AFAL TR-76-209-ADD-5



MDC-J7278-ADD-5

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DDC FILE COPY

PREPARED FOR
AIR FORCE AVIONICS LABORATORY
AFAL/AAA-1
WRIGHT-PATTERSON AFB, OHIO 45433

DTIC ELECTRONIC
APR 17 1980

410933 80 4 15 062

TABLE OF CONTENTS

<u>PARAGRAPH</u>		<u>PAGE</u>
1.0	INTRODUCTION	1
2.0	CONFIGURATION MANAGEMENT OVERVIEW	2
3.0	PROJECT ORGANIZATION	15
4.0	CONFIGURATION IDENTIFICATION	26
5.0	CONFIGURATION CONTROL	31
6.0	CONFIGURATION STATUS	42
7.0	SUBCONTRACTOR CONTROL	48
8.0	CONFIGURATION AUDITS AND REVIEWS	52
	APPENDIX A CHANGE CONTROL FORMS AND LOGS	61

Preceding Page BLANK - NOT FILMED

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

LIST OF FIGURES

<u>NUMBER</u>	<u>TITLE</u>	<u>PAGE</u>
1	SYSTEM DEVELOPMENT PROCESS	2
2	ACTIVITY SEQUENCE DURING CONCEPTUAL PHASE	6
3	ACTIVITY SEQUENCE DURING DEFINITION PHASE	7
4	ACTIVITY SEQUENCE DURING DEVELOPMENT PHASE	9
5	ACTIVITY SEQUENCE DURING INTEGRATION PHASE	11
6	ACTIVITY SEQUENCE DURING OPERATIONAL PHASE	13
7	RELATIONSHIP OF SOFTWARE PROJECT TO COMPANY STRUCTURE	15
8	WORK ORDER AUTHORIZATION	20
9	WORK RELEASE ORDER	21
10	ENGINEERING WORK ORDER - TYPE I	22
11	ENGINEERING WORK ORDER - TYPE II	23
12	SOFTWARE PROJECT MANAGEMENT ACTIVITY FLOW	25
13	SYSTEM LIFE CYCLE MODEL WITH COMPLETE DOCUMENTATION SET	27
14	CLASS I, II AND SUPPORT DOCUMENT CHANGE CONTROL FLOW CHART	36
15	ICWG INTERFACE CHANGE CONTROL FLOW CHART	39

LIST OF TABLES

<u>NUMBER</u>	<u>TITLE</u>	<u>PAGE</u>
1	MAJOR CONFIGURATION MANAGEMENT EVENTS	14

Accession For	
NTIS GRI&I	<input checked="" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	<input type="checkbox"/>
By _____	
Date _____	
Distribution _____	
Remarks _____	
Dist	_____
A	_____

1.0 INTRODUCTION

Purpose

This configuration management plan defines the policies and methods for implementation of configuration management for the IDAMST software.

Scope

The plan defines the basic functions of configuration management (CM) and the baselines or reference points for development and control of the IDAMST software documents and products. The major impact of configuration management is in the development phase of the program. Since program plans, schedules and contractual requirements for the development phase have not yet been formulated, this plan, of necessity, provides a general approach which will require modification and updating when this information is available. The major functions of configuration management covered by this plan are:

- a. Configuration Identification, The configuration of a computer program is identified by, and documented in, a series of specifications, some of which identify its required configuration and others its achieved configuration.
- b. Configuration Control, In the configuration control process, changes to the established specifications of a computer program and to the program itself are classified, evaluated, approved or disapproved, released, implemented, and verified. The purpose is to assure that the computer program configuration used in critical phases of testing, acceptance, and delivery is known and is compatible with the specifications.
- c. Configuration Status Accounting, Configuration status accounting is the recording and reporting of data concerning a computer program's configuration identification, proposed changes to its configuration identification, and the implementation status of approved changes.
- d. Verification, A series of configuration reviews and audits provide verification that the performance achieved by the product is the performance required by the development specification, and that the configuration of the product is accurately specified in the product specification.

2.0 CONFIGURATION MANAGEMENT OVERVIEW

Software management techniques for the IDAMST must be viewed against the background of the total STOL system development as shown in the simplified form in Figure 1.

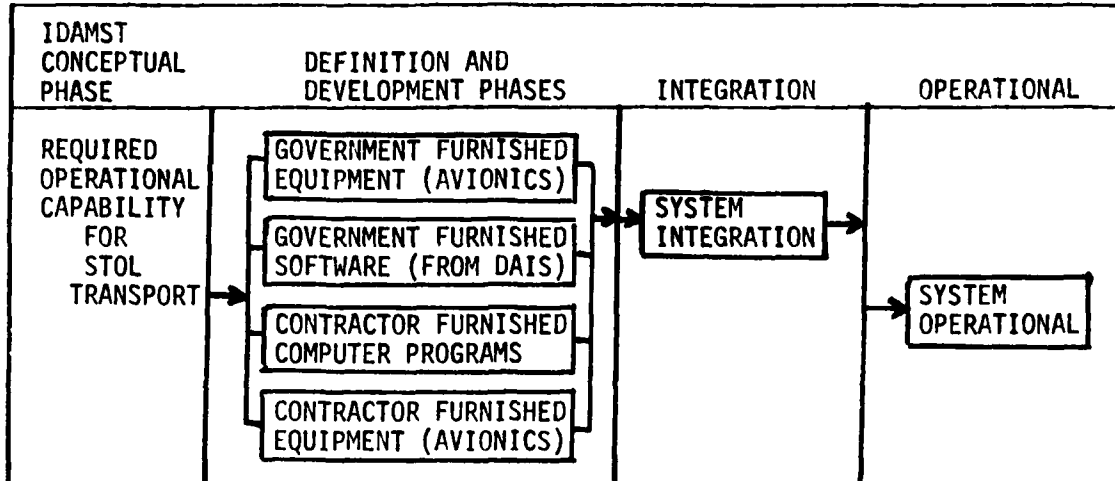


Figure 1. System Development Process

The Government Furnished Equipment (GFE) could be a computer, navigation and communication equipment, Government Furnished Software (GFS) from programs such as DAIS and a compiler. The hardware items are referred to as Configuration Items (CI's) and the computer programs are called computer program configuration items (CPCI's).

The phases of Configuration Management for a Government system as shown in Figure 2-1 are:

- a. Conceptual Phase. Determination of mission requirements and system requirements by the customer/user.
- b. Definition Phase. Definition by one or more contractors of the functional requirements of all major software and hardware elements of the system.
- c. Development Phase. Preliminary analysis and design, detailed analysis and design, development (for computer programs, consists of coding, debugging, and checkout), development testing, and validation testing of system elements.

- d. Integration Phase. Installation, integration, and test of system elements in the operational environment by the development contractor(s) and sometimes an independent integration contractor.
- e. Operational Phase. Operation, maintenance, and product improvement by the customer/user, development contractor(s), and integration contractor if there is one.

The characteristics of an evolving system and its configuration items are defined and documented in increasing detail at logical transition points, or baselines, of the system life cycle.

Three baselines are defined in the acquisition phase as:

- a. Functional baseline. Marks end of conceptual phase and start of definition phase. Is established by System Specification.
- b. Allocated baseline. Marks end of definition phase and start of development phase. Is established by development specification.
- c. Product baseline. Marks end of development phase and start of integration phase. Is established by product specification.

Later in the life cycle, a fourth baseline occurs:

- d. Operational baseline. Marks end of integration phase and start of operational phase. Is established by satisfactory demonstration of entire system in operational environment.

Major events within the development phase, such as beginning of coding, formal establishment of interface requirements, and internal delivery of the product to an independent testing team, sometimes are given the status of baselines and are called "internal milestones."

From among the various terms used by the Air Force to describe configuration management elements, this plan has adopted a standard set of terms that offers a common basis for understanding the principles discussed. This standard set includes the life cycle phases, the baselines, the baseline specifications and other documents, and the levels of the computer program hierarchy.

The system, development, and product specifications are the principal documents of baseline configuration management. The complete software documentation set described in this plan is as follows:

- a. System specification
- b. Development specification
- c. Product specification
- d. Interface specification
- e. Data requirements specification
- f. Validation test plan
- g. Validation test procedure
- h. Computer operators manual
- i. Program maintenance manual
- j. Data base document
- k. Development test plan
- l. Development test procedure
- m. Programmer notebook
- n. Version description document

COMPUTER PROGRAM HIERARCHY

The complete hierarchy of computer program levels adopted for this plan is as follows:

- a. Software Segment. A package of related computer programs contracted to one contractor.
- b. Computer Program. A computer program or portion of a computer program that satisfies an end use function and is contractually designated for configuration management. Contractually, a computer program in this sense is called a "computer program configuration item" (CPCI).
- c. Function. A functionally or logically distinct part of a computer program. Contractually, a function is called a "computer program component" (CPC).
- d. Module. An optional level consisting of two or more routines.
- e. Routine. By definition, the lowest compilable unit of a computer program.

SOFTWARE DEVELOPMENT UNDER BASELINE CONFIGURATION MANAGEMENT

Conceptual Phase

The sequence of activities during the conceptual phase, diagrammed in Figure 2, is as follows:

1. Identify Operational Requirements. Analyze mission requirements and define operating concept, modes, environments, and constraints.
2. Define Initial System Concept. Define qualitative requirements to establish potential value of system and interest in it.
3. Conduct System Feasibility Studies. Formulate basic system requirements, considering timeliness, technological and economic feasibility, and capability. Identify major risks and recommend later resolution.
4. Perform System Engineering. Define system requirements further, including installation with other systems, concept of operation in mission/environmental context, allocation of functions between man and machine, and performance characteristics of the computer.

5. Determine Design Requirements and Gross System Functions. Define essential functional interfaces. Define requirements for overall system performance and system testing. Identify all major system elements to be developed or otherwise acquired.

The major product of the conceptual phases is the system specification. Its formal acceptance at the System Requirements Review (SRR) establishes the functional baseline.

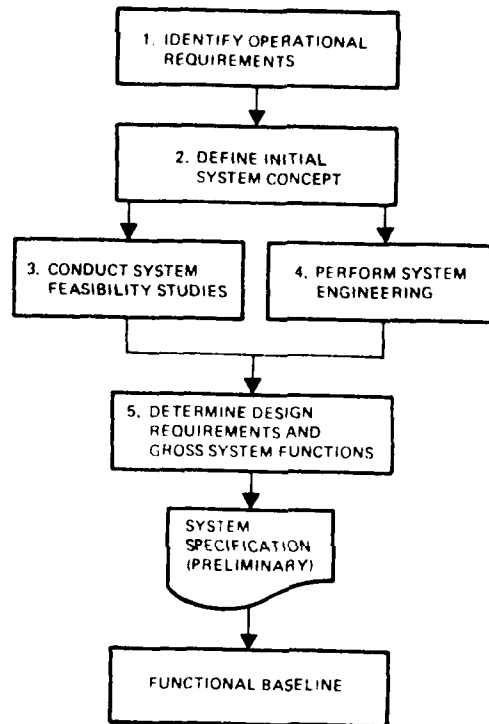


Figure 2. Activity Sequence During Conceptual Phase

Definition Phase

The sequence of activities during the definition phase, diagrammed in Figure 3, is as follows, with numbering continued from the preceding subsection:

6. Develop Interface Control Requirements. Define interface between operational functions. Include (a) initial performance allocation by segment, (b) schedule, and (c) control techniques.

7. Expand System Requirements. Perform a comprehensive and critical review of functions, performance, and design requirements and expand them.
8. Define Computer Program Requirements. Define requirements for the detailed tasks to be performed by the operational computer programs.
9. Define Manual Tasks. Define manual tasks and procedures, including console operator procedures and decisions, and the design requirements for console controls and displays.
10. Define Equipment Requirements. Define requirements for the detailed tasks to be performed by the computer and other system hardware.
11. Prepare CPCI Development Requirements. Specify the general requirements for the design, development, test, and validation of the CPCI's.

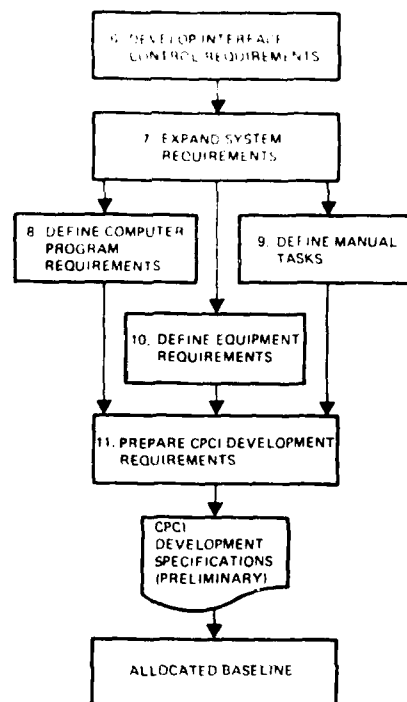


Figure 3. Activity Sequence During Definition Phase

The major products of the definition phase are the CPCI development specifications. Their formal acceptance by the customer at the System Design Review (SDR) establishes the allocated baseline.

DEVELOPMENT PHASE

During the development phase, the sequence of activities, diagrammed in Figure 4, is as follows, with numbering again continued:

12. Review CPCI Development Specifications. Perform a comprehensive and critical review of the CPCI development specifications. Both the customer and the development contractor must accept these specifications with their updated costs and schedules at the PDR.
13. Preliminary Design (CPCI's). Complete system input/output/function allocation; segment programming tasks into specific packages by function, module, or routine; and develop refined functional flows. If possible, use pilot team to design, code, and validate critical areas of the design, and use a system simulation to study timing, accuracy, storage, etc., to avoid risk of serious design deficiencies.
14. Preliminary Design Review (PDR). Conduct joint customer/contractor review to confirm design integrity. Contractor presents results of preliminary design, including pilot team effort.
15. Detailed Design (CPC's). Prepare flow charts, logic diagrams, equations, and narrative description, sufficiently detailed to provide basis for actual coding. Complete definition of data base at the software segment, program, function, module, and routing levels, including number, type, and structure of tables and description of items in the tables. Define all elements as critical or noncritical.
16. Validation Test Plan. Develop detailed draft of validation test plan for review by customer at CDR.
17. Critical Design Review (CDR). Conduct joint customer/contractor review to confirm that design meets its development requirements and is defined sufficiently to permit start of coding.
18. Coding, Debugging, and Checkout. Code, debug, and check out routines. Debugging will ensure compilation, and checkout will ensure execution. Output data itself is not validated until development testing.
19. Development Testing. Conduct development testing of coded elements up to CPCI level, then integrate software segment to produce preliminary master tape. Development testing is performed by the development programmers.

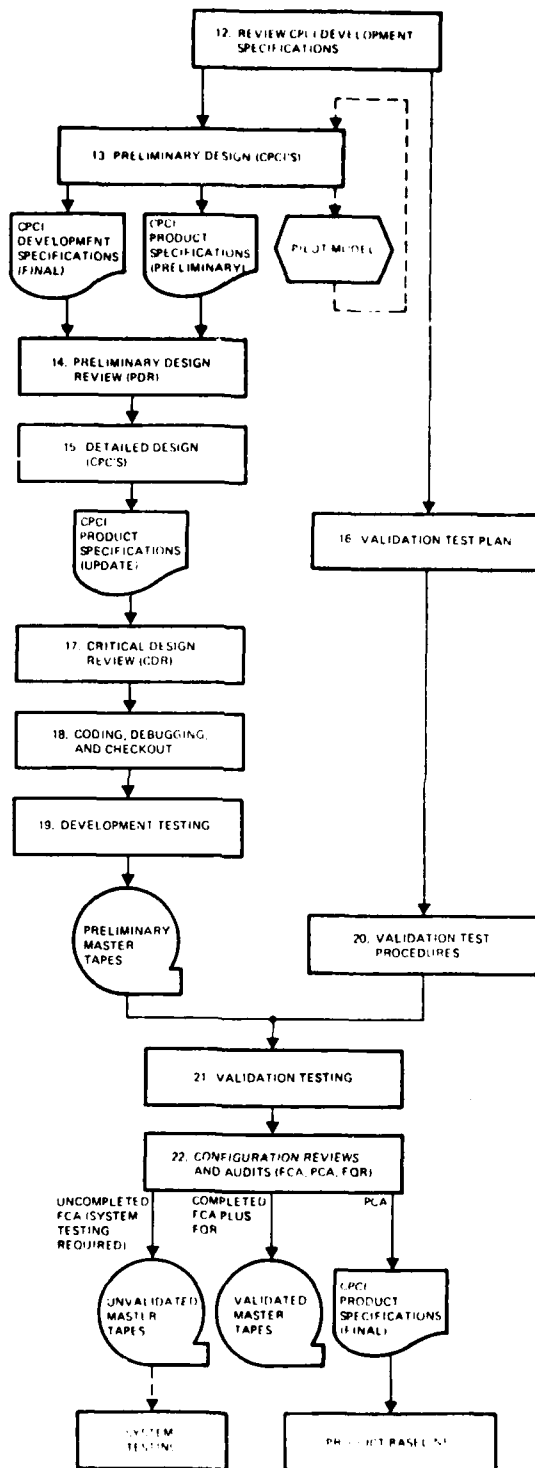


Figure 4. Activity Sequence During Development Phase

20. Validation Test Procedures. Prepare detailed test procedures for validation of software and submit to customer for approval.
21. Validation Testing. Conduct two-stage validation testing of CPCI's using independent test group. In first stage, perform in-house validation testing of CPCI's against requirements of development specifications, correct problems, and update documentation. In second stage, rerun validation tests for customer or review results of successful in-house validation test runs with customer. In addition, validate product specifications and other relevant documentation in both stages.
22. Configuration Reviews and Audits (FCA, PCA, FQR). Conduct Functional Configuration Audits (FCAs), Physical Configuration Audits (PCAs), and Formal Qualification Reviews (FQRs) as required to obtain customer acceptance of software products and documentation.
 - o An FCA verifies that the performance achieved by the product during validation testing is the performance required by the development specification. If integrated system testing is necessary to fully verify a product, the FCA is not completed until that time.
 - o A PCA verifies that the configuration achieved in the product is accurately specified in the product specification and thereby establishes the product baseline. For Government contracts, a DD Form 250, "Material Inspection and Receiving Report," is signed by the customer for each CPCI successfully audited. Preliminary reviews of computer operators manuals, program maintenance manuals, and any similar document also are conducted at a PCA; formal acceptance of these manuals usually is withheld until system testing.
 - o An FQR, like an FCA, verifies that the actual performance of a product, as determined through validation testing, is the performance required by the development specification. In addition, it verifies incorporation of changes to the product and documentation that may have been required by the FCA and PCA, identifies the product's validation test data, and is the customer's final certification of the product. When sufficient validation test data is available at FCA to assure the product's performance in its system environment, the FCA and FQR are conducted at the same time.

The major products of the development phase are the CPCI's (some completely validated, some not) and their final product specifications and other documentation. Formal acceptance of a product specification at a PCA establishes the product baseline.

INTEGRATION PHASE

During the integration phase, the sequence of activities, diagrammed in Figure 5, is as follows:

23. System Testing. Conduct testing of entire integrated electronic data processing (EDP) system, including the CPCIs developed and tested during the development phase and other interfacing system hardware and software to ensure that the EDP system satisfies the performance and design requirements of the applicable system specification.
24. Operational Testing. Test integrated and validated EDP system in the total operational environment with all other associated systems.

During the integration phase, FCAs and FQRs are conducted for CPCIs that could not be fully validated previously because of the need for more realistic operating conditions. Successful demonstration of minimum operational capability of the entire system establishes the operational baseline.

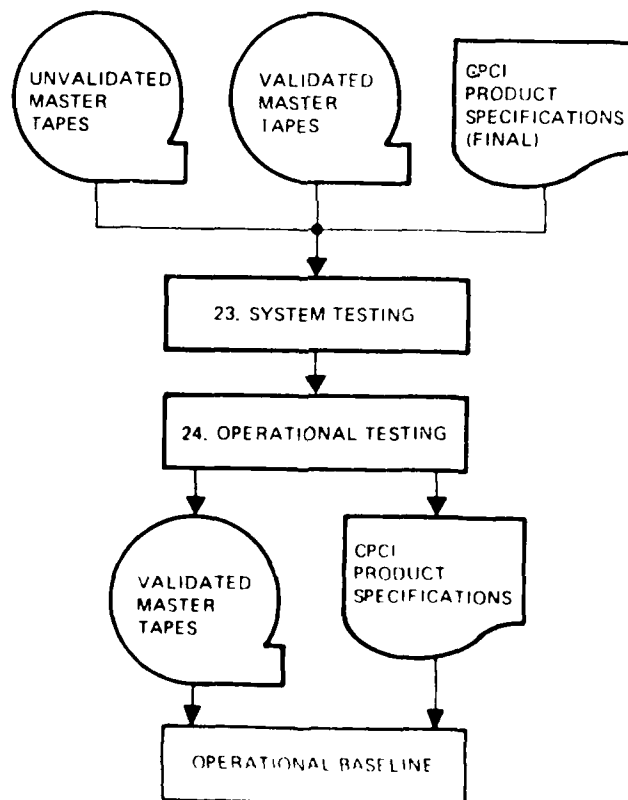


Figure 5. Activity Sequence During Integration Phase

OPERATIONAL PHASE

The sequence of activities during the operational phase, diagrammed in Figure 6, is as follows:

25. Turnover to User. Turn computer programs and user documentation over to the using organization. Put entire EDP system under customer configuration control.
26. Update of Changes on Contract. Contract, test, and record system deficiency corrections and/or improvements, using the configuration management system.
27. Transfer of Engineering and Logistics Responsibilities. Transfer responsibilities for controlling corrective actions and minor improvements to the organization that will operate the EDP system. These responsibilities include maintenance of computer operators manuals, program maintenance manuals, and other software documentation.
28. System Update. Update system specification to incorporate changes resulting from configuration management.
29. System Operation on Routine Basis. Operate, maintain, and refine the EDP system and its role through continuing routine procedures. Incorporate product improvements or use them as the basis for the next generation system.

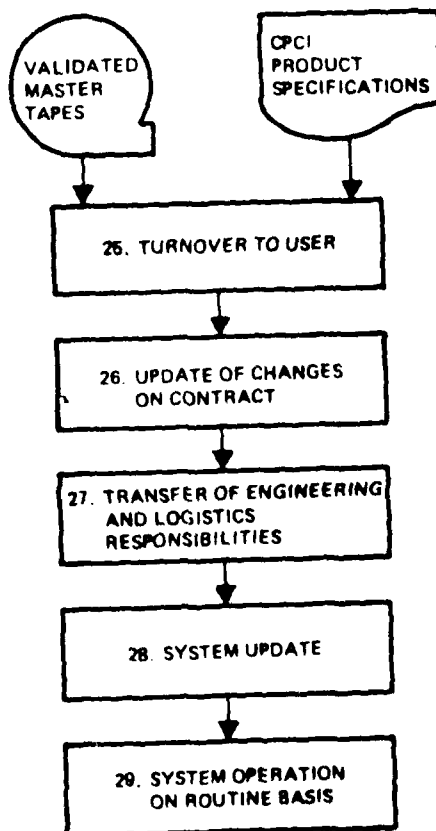


Figure 6. Activity Sequence During Operational Phase

TABLE 1. MAJOR CONFIGURATION MANAGEMENT EVENTS

The Sequence of Major Configuration management Events occurring during a Medium-to-Large Software Development Program is shown.

		Definition Phase		Development Phase			
		Configuration Management Events					
Activities		At Functional Baseline	At Allocated Baseline	At PDR (Preliminary Design Review)	At CDR (Critical Design Review)	At Internal Delivery	At Product Baseline
Configuration Management Planning	Product Development		Preliminary design starts	Detailed design starts	Coding starts	Turnover to independent test group	Customer accepts product
	Configuration Management (CM) Plan		Customer approves preliminary CM plan	Customer approves updated CM plan			
	Configuration Management (CM) Procedures		DAC starts preparation of detailed CM procedures	DAC modifies CM procedures according to updated CM plan	DAC completes detailed CM procedures		
Configuration Identification	Functional Baseline Identification (System Spec)	Customer releases preliminary system spec	Customer releases final system spec				
	Allocated Baseline Identification (Development Specs)		Customer approves preliminary development specs	Customer approves final development specs			
	Product Baseline Identification (Product Specs)			Customer approves preliminary product specs	Customer approves updated product specs		Customer approves final product specs
Configuration Control	Project Change Control Board (CCB)		DAC establishes project CCB for processing Class I Changes				
	Class I and II Change Control	Customer initiates change control of functional baseline	Customer and DAC initiate change control of allocated baseline				For OS&M contracts DAC initiates change control of product baseline
	Internal Change Control				DAC initiates internal change control of updated product specs	DAC initiates internal change control of product*	
Subcontractor Control	Status Accounting	Customer initiates status accounting of functional baseline	Customer and DAC initiate status accounting of allocated baseline		DAC initiates status accounting for updated product specs	DAC initiates status accounting for product*	For OS&M contracts DAC initiates status accounting for product baseline
	Subcontractor Control			DAC initiates subcontractor control if required			

* For minimum capability routines released during development testing, DAC initiates internal change control and status accounting upon release

BEST PAGE IS BEST QUALITY PRACTICABLE
 FROM COPY AVAILABLE TO DDC

MCDONNELL DOUGLAS CORPORATION

3.0 PROJECT ORGANIZATION:

A typical project organization for software management is shown in Figure 7.

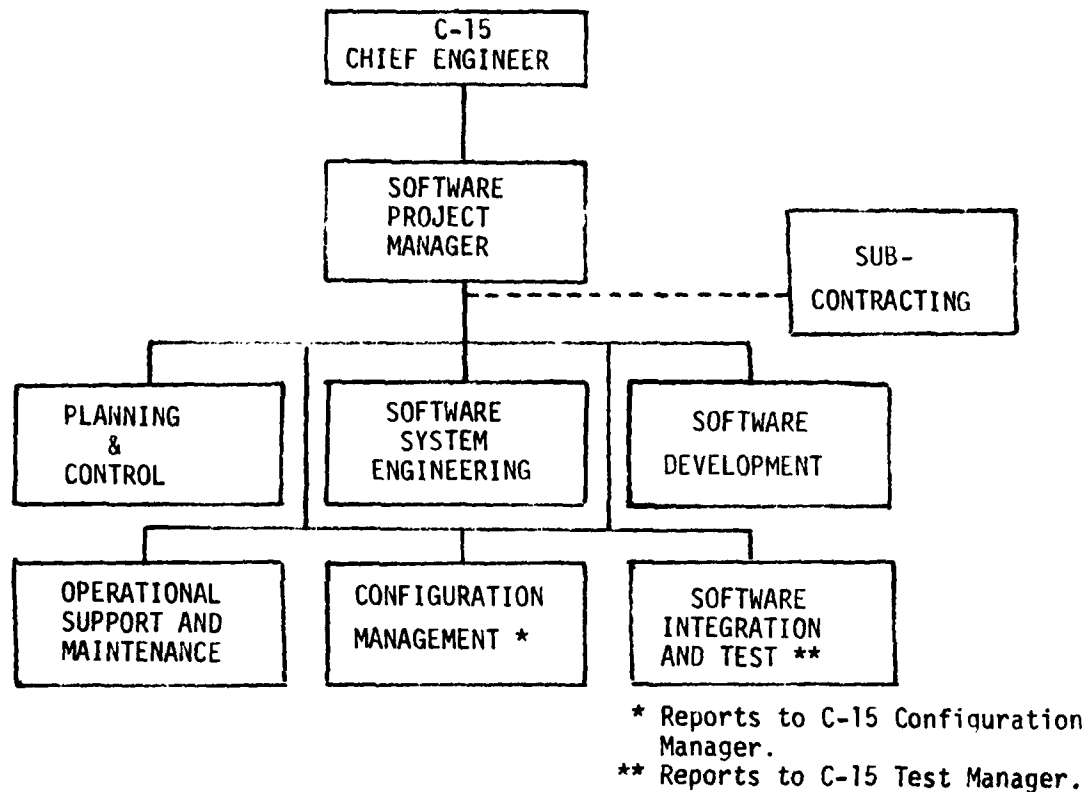


Figure 7. Relationship of Software Project to Company Structure

The project management structure is sized for a peak manloading of approximately forty persons. The IDAMST complexities will require a software group to cover the many facets identified above. A single contract for both development and operational support and maintenance is assumed. The influence of project size, end product, schedule, special customer requirements and other project - peculiar factors must receive special attention.

PROJECT MANAGEMENT FUNCTIONS

Project Manager

Configuration management policy for a software project covers baseline documentation, configuration change control, configuration status accounting, software release and storage, reviews and audits, and subcontract configuration management.

Well in advance of contract start, the project manager plans the project organization, allocates management responsibilities, including CM responsibilities, and sets the overall CM policy.

Before the start of a project and during the early phases, the project manager's primary task is planning major functions of the project and ensuring that appropriate planning is being conducted at lower levels within the project. One product of this early period is an overall project implementation plan. Two others are project-specific CM and documentation plans. The project manager has a vital interest in seeing that these plans are prepared and that they accurately reflect the constraints within which he must operate and the policies he wishes to implement.

As a project matures, the project manager's emphasis shifts to monitoring and controlling, although plans must be continually extended, updated, and revised to remain credible and useful. The project manager and his assistants must measure actual technical performance and adherence to schedule and cost against project plans, and take corrective actions when actual performance, schedule, or cost show signs of departing from the plans. CM supports this monitoring function by establishing requirements for technical documentation and establishing configuration reviews and audits. The documentation, reviews, and audits help provide the visibility that the project manager and his APM's require to monitor and measure technical performance against schedule.

Configuration Management Manager

The CM manager will be responsible for:

- a) Preparation of the CM plan and the detail procedures necessary to implement this plan.
- b) Implementation and phasing of the CM plan and related procedures.

The CM manager will be responsible for the organization, preparation, and implementation of policies and procedures required to perform the level of CM required by the project, as approved by the project manager.

The CM manager for Software will report to the C-15 System Configuration Manager.

Software Integration and Test Manager

The primary task of the software integration and test manager is to verify that the software product conforms to performance and design requirements of the applicable specifications. The Software Integration and Test Manager will report to the C-15 System Test Manager.

The software integration and test manager is responsible for:

- a. Organization and supervision of the integration and test group.
- b. Providing input for test sections of development and product specifications.
- c. Preparation of validation test plans and test procedures.
- d. Performing validation testing (includes software segment testing and acceptance testing).
- e. Supporting development testing, system testing, and operational testing.

System Engineering Manager

The system engineering manager contributes to CM activities by:

- a. Generating the development specifications, interface specifications, data requirements specifications, and preliminary product specifications, and maintaining the first three of these.
- b. Monitoring internal and external functional and physical interfaces to ensure compliance with applicable specifications and approved changes thereto.
- c. Monitoring development progress to ensure compliance with technical performance requirements.
- d. Providing a member of the project Change Control Board.
- e. Providing a project representative to the customer's Change Control Board as required.
- f. Providing a member of the Interface Control Working Group.

Planning and Control Manager

The planning and control manager contributes to CM activities in three ways:

- a. Assists the project manager in planning, scheduling, and monitoring project design reviews.
- b. Correlates configured computer programs and functions, modules, and routines thereof with WBS elements and with budgets by means of EWO's thereby providing traceability from the various structural levels of the delivered end product to responsible organizational units within the project and to the allocated budgets.
- c. Manages the Data Management Office. This office ensures that all contractually required data are generated on time and are consistent with project standards. It produces the documentation plan that defines the format and content of every deliverable data item.

Software Development Manager

The software development manager is responsible for the design, coding, and development testing of his software products. He also is responsible for generating materials for configuration management events, including reviews and audits, and test documentation for which no external approval is required. His organization updates the preliminary product specifications and writes the final product specifications.

Operational Support and Maintenance Manager

The manager for operational support and maintenance (OS&M) should be identified and his responsibilities should begin well before the end of development testing. This allows his organization the time to write any computer operators manuals or program maintenance manuals that are required for OS&M, and to validate such manuals during development testing, validation testing, and system testing. Other responsibilities include reviewing the Interface Control Document (ICD) and writing an OS&M plan.

After validation testing, the OS&M manager supports system testing. With the start of operational testing, he becomes the principal submanager on the project. His chief functions then are to support operational testing and use of the software and to maintain the software. These functions are performed by an OS&M team together with system engineering personnel under the OS&M manager.

Subcontract Administration

The subcontract administrator is the main interface with the subcontractor on all contractual matters. He maintains the subcontract, and his signature is required on all changes affecting the subcontractor.

Project Work Authorization

Upon receipt of Authorization to Proceed (ATP) from the customer, DAC Contracts issues a Work Order Authorization (WOA) (Figure 8) which is the primary document in the work authorization system.

The secondary authorizing document is the Work Release Order (WRO) (Figure 9) which is used to release work to the performing organization. WRO's can only be issued on the authority of a primary authorization document and assign the shop order number against which cost are collected and provide (either directly or by reference) the total budget, master schedule, and task definitions for all work authorized.

The Program Manager, upon receipt of the WRO, will release Engineering Work Orders (EWO's) which provide detailed task descriptions, budgets, and schedules for specific functions. The Type I EWO (Figure 10) is the initial EWO on a program and changes are made by the Type II EWO (Figure 11).

Complete traceability is maintained for all work authorization documents. Each secondary and detailed authorizing document references its authority document. In addition, each issuing organization maintains a cross reference file listing all lower level authorizations issued for each authority received.

WORK ORDER AUTHORIZATION

OS-100 (11-67)

TO

WOA NO.

SUBJECT

DATE

AUTH

ECP. NO.

QUOTATION No.

OFFICE ADMINISTERING CONTRACT

INSPECTION OFFICE

AGENCY PLACING ORDER

REQUISITION No.

APPROPRIATION

SHOP ORDER

CONTRACT No.

MODEL No.

CUSTOMER SERIAL No.

FIGURE 8.

WORK RELEASE ORDER DAC 30-110AB (REV. 1-72)		DOUGLAS AIRCRAFT COMPANY	DATE	W. R. O. NUMBER
				CONTRACT IDENTITY
SUBJECT				PROGRAM NAME & MODEL
				CUSTOMER
W. R. O. / E. W. O. NUMBER	ECP NO.	CPI NO.	SCN NO.	WORK BREAKDOWN IDENTITY NUMBER
SCHEDULE IN FOR.	PRODUCTION EFFECTIVITY			SHOP ORDER NO.
				PRIORITY RATING
INT. RELEASE DATE	SERVICE CHANGE NO.	SERVICE CHANGE EFFECTIVITY		
REVISION:				
REFERENCE:				BUDGET NUMBER

FIGURE 9.

APPROVALS				
COPIES		CONTRACTS		PROGRAM MANAGER
ENGINEERING ONLY CHARGE TIME TO →	SALES ORDER	FOR FURTHER INFORMATION REGARDING THIS WORK RELEASE ORDER, CONTACT (NAME) (PHONE)		ISSUING AUTHORITY
	E. W. O. NUMBER			

MODEL VER	TITLE	SOURCE INFORMATION GROUP CODE		ENGINEER	DATE
DEVELOP REQD	SUPPLIER REF DOCUMENT	PERCENT	DATE		
YES <input type="checkbox"/> NO <input type="checkbox"/>					
GENERAL WORK OUTLINE & REASON:					

COST CHARGE NO	BASIC EWO	DASH	GRP	OP	HOURS	START DATE	DUE FROM GRP	SYS	BY	DEPT	TITLE OF DETAIL JOB

EFFECTIVITY	DPLT	HOURS	TOTAL ESTIMATE			CONTROL USE ONLY			LEAD CONTROL
			NO	EXPS AUTH	NO	HOURS	OBJECT	EXP ACCT	

APPROVALS

CGGN ZANT ENGINEER _____

PROJECT OFFICE _____

FIGURE NO. _____

Software Project Activity Flow

A top level flow of project activities during the lifetime of a software project is shown in Figure 12. The emphasis is on activities related to CM or otherwise essential to successful execution of the project. The heavy arrows form the critical path for most projects. This path connects the dependent activities that are most time critical and therefore determines the overall schedule of the project.

In Figure 12, the contract period is assumed to include the development phase, the integration phase, and at least part of the operational phase of the system life cycle. However, a software contract may start and stop at any point in a system's life cycle. The Air Force usually procures software design and development under a "development contract" and software operational support under an "operational support and maintenance (OS&M) contract."

The contract period in Figure 12 is divided into three parts for convenience of reference:

- a. Preliminary design (from contract start to PDR).
- b. Development and test (from PDR to product baseline).
- c. Operational support and maintenance (OS&M) (from start of system testing to end of contract).

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

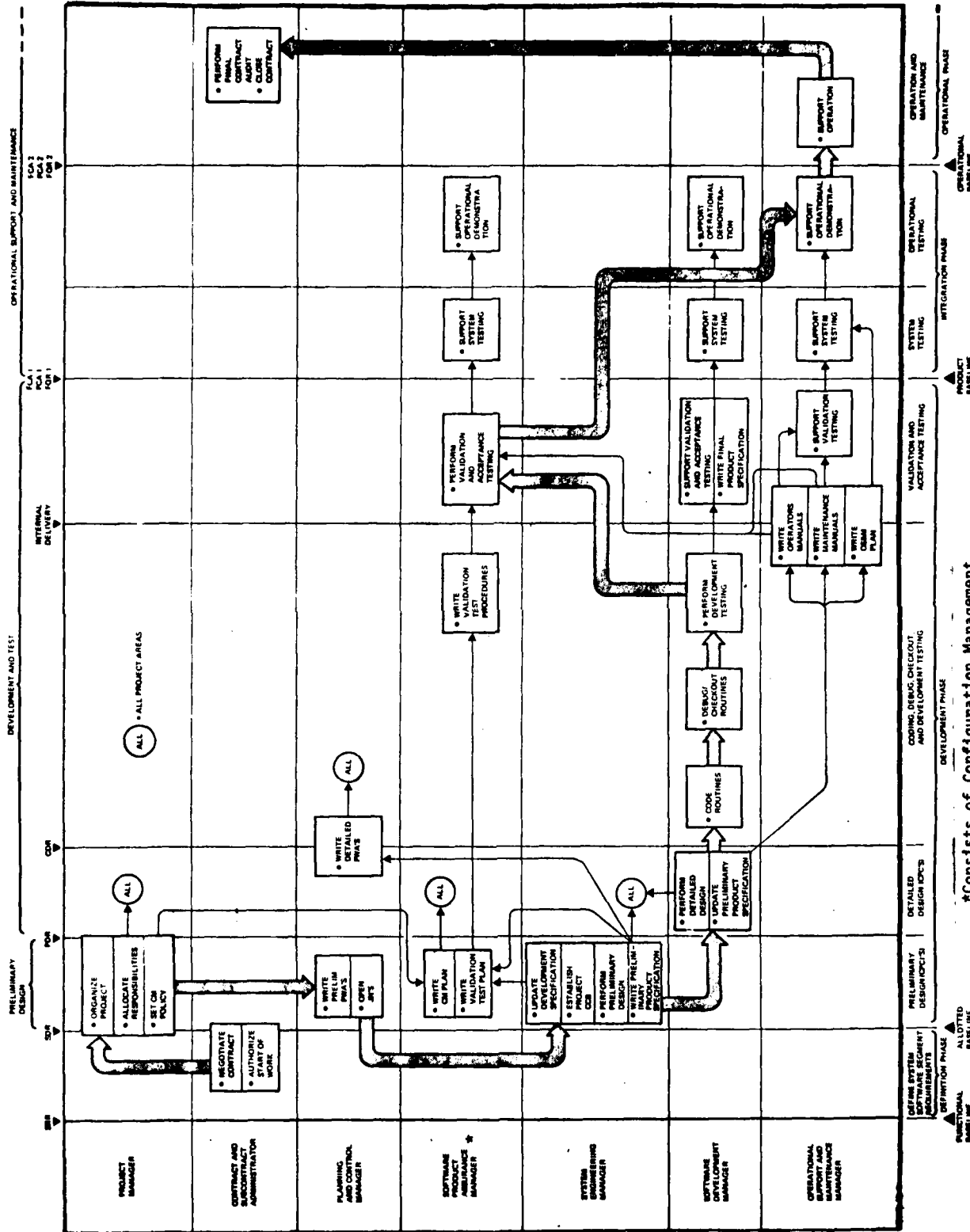


Figure 12. Software Project Management Activity Flow

4.0 CONFIGURATION IDENTIFICATION

Configuration Management is a progressive and increasingly detailed identification of software and hardware configuration by means of baseline documentation Figure 13. The configuration of a software or hardware item is identified by, and documented in a series of specifications which identify its required configuration (development specifications) and its achieved configuration (product specifications). This identification is the basis for configuration control, status accounting, and reviews and audits as well as other program documentation.

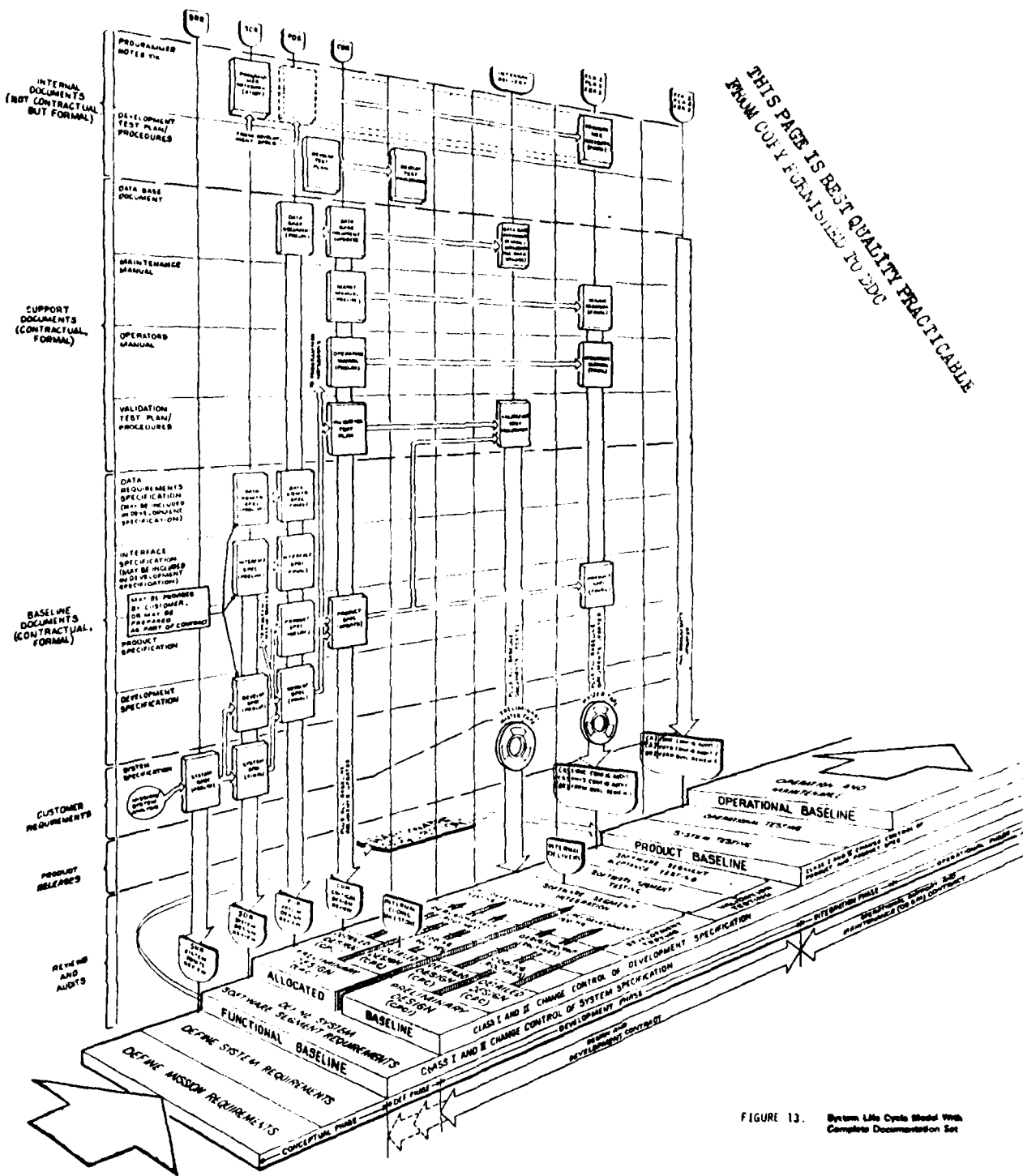
Support documents that are not part of the baseline specification, Fig. 13 are influenced by, but do not identify or control the hardware or software configuration identification. These two groups of documents require different change control procedures.

Documentation

Baseline Specification Documents

The baseline documents are formal contractual deliverables or governmental agencies specifications that establish and identify the baselines for each hardware or software product. The type of specifications for the IDAMST Program are identified below. MIL-STD 483 and MIL-STD 490 contain the guidelines for these specifications.

<u>Specification</u>	<u>Type</u>
System Specification	A
System Segment Specification	A
Prime Item Development Specification	B1
Prime Item Product Specification	C1
Critical Item Development Specification	B2
Critical Item Product Specification	C2
Computer Program Development Specification	B5
Computer Program Product Specification	C5
Interface Specification (or Interface Control Drawing)	
Data Requirements Specification	



THIS PAGE IS BEST QUALITY PRACTICABLE FROM COPY FURNISHED TO SDC

FIGURE 13. System Life Cycle Model With Complete Documentation Set

MCDONNELL DOUGLAS CORPORATION

The specific IDAMST specifications are defined in the specification trees contained in the IDAMST System and System Segment Specification TBD. The system, development and product specification shall be prepared in accordance with MIL-STD 490.

The Interface Specification or Interface Control Drawings shall bring together into a single document all available data on interfaces.

The Data Requirements Specification is to list and define data elements which the system must handle and communicate data collection requirements to the user.

CPCI Breakdown

A CPCI breakdown is a description of the software elements at all levels of a CPCI structure. This will be determined by the allocation of CPCI requirements down to the lowest compilable level. The hierarchy of software program levels for IDAMST is as follows:

a. Software Segment

All CPCI's developed by one contractor or one government agency

b. Software Program (CPCI)

A software program or portion of a software program that satisfies an end use function and is designated for configuration management as defined in the IDAMST Specification Tree. Contractually, a software program in this sense is called a "Computer Program Configuration Item" (CPCI).

c. Program Module

A functionally or logically distinct part of a software program (CPCI). Contractually, a function is called a "Computer Program Component" (CPC).

d. Tasks

e. Comsubs

Support Documents

Support documents are derived from the baseline specification documents to provide design, test, and demonstration information required for the IDAMST project. The support documents are formal contractual deliverables or government agencies documents which are identified below:

a. Software Support Documents

o Data Base Documents

- o Validation Test Plan/Procedures
- o Validation Test Reports
- o Computer Program Operation Manual
- o Listings

b. System Integration, Test and Documentation Support Documents

- o Demonstration and Acceptance Plan
- o General Test Plans/Procedures
- o General Test Reports

Software Products

The software products to be developed for IDAMST will be defined by the System and Development Specifications. The specific products for IDAMST include:

- a. Software Products (Software Program Physical Media)
 - o Magnetic Tapes
 - o Disk Packs
 - o Card Decks

Commercial support software will not be placed under Configuration Management.

Numbering of Products and Documents

A suitable numbering system for the software products and documents is required to achieve configuration traceability of these elements throughout the program's life cycle. The following items of configuration identification shall be included in a number system:

- a. Program Plans and Content

Plans, operating instructions (OI's) and data (e.g., Work Breakdown, Structure, Master Schedules, System Schedules, etc.)
- b. Documentation

Specifications and support documents
- c. Physical Configuration Item

Physical hardware and recording media for the software programs: magnetic tapes

d. Change Control Documentation

Forms used to initiate and control configuration changes.

Program Plans and Content Definition

Each Program Plan, Work Breakdown Structure and Schedule shall have a unique reference number for controlling and reporting configuration. The reference number shall contain: (TBD)

Documentation Identification

Each specification and support document shall have a unique reference number for controlling and reporting configuration. The reference number shall contain: (TBD)

CPCI Level Identification

Program modules, tasks, and comsubs are the lowest compilable elements of a CPCI. Each contractor or government agency responsible for development of a CPCI shall apply configuration control at the program module, task and com-sub level. The change status of these levels shall be tiered up to the CPCI level, which is the level reported to IDAMST Configuration Control Board (CCB). Traceability of these levels by a tape/disk library and log shall be accomplished by the contractor or government agency.

Physical Configuration Item Identification

Each physical CI shall have a unique reference number for controlling and reporting configuration status. The physical hardware number shall be in accordance with (TBD).

Since magnetic tapes usually contain more than one configuration controlled CPCI, a tape cannot be identified by a CPCI identification number alone. Instead, a fixed number shall be assigned to each tape, with a prefix specifying the type of tape (source, object, etc.) and a suffix specifying the CPCI as follows: (TBD)

Change Control Form Identification

Change control numbers are required on all configuration management forms defined in Section 6. A prefix shall be assigned to indicate type of form and numbers assigned sequentially to the form as they are initiated as follows: (TBD)

5.0 CONFIGURATION CONTROL

Change Control

After each software configuration is formally established at a baseline, each proposed change shall be carefully controlled to ensure adequate review and evaluation of change effects prior to approval or disapproval. Configuration control is the process by which changes to the software documents and products are initiated, classified, evaluated, approved or disapproved, released, implemented and verified. This process assures adequate analysis of changes that effect cost, schedule, and performance; and provides controlled updating of the evolving software and hardware configuration throughout the IDAMST project.

Change Classification

Changes to the IDAMST software and hardware products or documentation shall be placed into three classifications, each requiring different processing and reviewing procedures, depending on the impact of the change:

a. Class I Changes

Class I changes require review and approval by the AFAL IDAMST Configuration Control Board (CCB) prior to implementation.

b. Class II Changes

Class II changes do not require IDAMST CCB approval prior to implementation, but copies of Class II changes must be forwarded to IDAMST CCB for concurrence of the change classification.

c. Internal Changes

Internal contractor or other governmental changes do not require IDAMST CCB approval or concurrence.

Class I Changes

Class I changes are major technical or non-technical alterations to a software product, or to baseline specifications. The Class I changes shall include the following kinds of changes:

- a. Technical changes to the functional or allocated configuration identification of the configuration item.
- b. Technical changes to the product configuration identification.
- c. Changes involving performance outside the specified tolerances of the product configuration identification or affecting interface characteristics of the product configuration.

- d. Changes affecting non-technical provisions of the contract, such as fees, incentives, cost, schedules, guarantees, or deliveries.
- e. Changes affecting government furnished equipment (GFE) or government furnished software (GFS), computer programs, safety, support equipment, training equipment, delivered support documents whose revision is not funded by the contract.
- f. Changes that warrant assignment for a new identification number to the configuration item.

Changes to correct a deficiency that prevents the product from performing with contractual requirements are called "compatibility changes." Class I changes resulting from new or modified requirements outside the scope of the existing contract are called "design changes."

Class II Changes

Class II changes are those that do not fall within the definition of a Class I change. Generally they are minor non-technical changes affecting documentation only, such as correction of documentation errors, addition or supplementary information, or changes defined as minor by AFAL contract monitor.

Internal Changes

Internal changes are changes to a configuration item element, or document that is under contractor or agency internal change control only and has not yet been released to IDAMST Program CCB for Class I and Class II change control.

Change Control Phasing

Configuration change control is primarily concerned with the development phase as shown in Figure 4.1. This figure shows the effective periods of change control in relation to the product, specifications, and support documents. At the product baseline, all products and documentation are under Class I and II change control and continue throughout the remaining IDAMST system cycle.

Configuration Control Board (CCB)

The AFAL IDAMST Configuration Control Board (CCB) shall be responsible for reviewing, evaluating, approving or disapproving, and release of Class I changes; concurring in the classification of Class II changes; and concurring and releasing supporting documentation.

The primary CCB members shall be:

1. IDAMST Chief Engineer (CCB Chairman)
2. Program Control Manager (CCB Secretary)
3. AFAL Integration Manager
4. AFAL Hardware Manager
5. AFAL Software Manager
6. AFAL Facilities Manager
7. Representative from IDAMST software contractor
8. Representatives from other government agencies

Each primary board member will select an alternate board member to act in his absence. At the discretion of the CCB Chairman and as required in the interests of the IDAMST Program, agencies other than those identified shall be requested to participate in CCB activity as required.

Each member of the CCB makes recommendations for approval or disapproval and the chairman makes the final approval or disapproval decision.

Change Control Forms

Change control forms are the basic media for initiating, evaluating, approving or disapproving, releasing and implementing changes. These forms are to be used for reporting problems, requesting modifications, and submitting change proposals. The IDAMST configuration change control system will be implemented with the following set of forms:

a. Problem Report (PR)

The PR is used to report a known or suspected problem in software or hardware design, documentation, or interface; schedule incompatibilities, test anomalies, et al. The PR provides a record of the problem and is used to notify the appropriate organization or contractor for resolution. The PR also is used to transmit the closing action on the problem.

b. Data Base Change Request (DBCR)

The DBCR is the primary vehicle for coordinating data base changes. A DBCR is required for every change to a data base after the data base is placed under configuration control.

When a problem is reported via a PR and the solution to the problem requires a data base change, a DBCR is prepared and distributed. After approval of the DBCR, the data base is updated accordingly.

c. Engineering Change Proposal (ECP)

The ECP is used to propose Class I changes to IDAMST CCB. It describes the merits of the proposed change, the desirability of authorizing the required funding, and the available alternatives, giving the AFAL the information needed to evaluate engineering changes that have contract cost and schedule implications.

d. Specification Change Notice (SCN)

The SCN is used to propose, transmit, and record changes to a specification. Initially, it is used to submit Class I Specification change pages (accompanied by an ECP) for AFAL approval. After a proposed documentation change is approved, the SCN is used to transmit the change pages to document holders.

e. Configuration Control Board Directive (CCBD)

The CCBD is used to record and transmit CCB decisions on ECP/SCN changes to the procuring organization to initiate contractual authorization by a CCN.

f. Contract Change Notice (CCN)

The CCN is used to prepare, transmit, and record changes to a contract as directed by the CCBD. It is used to define the scope of the change.

g. Document Update Transmittal (DUT)

The DUT is used to transmit a draft copy of baseline and support documentation change pages to the IDAMST Program Control Manager. It tracks and controls a documentation change from the time the change is originated until it is officially concurred by IDAMST CCB for incorporation into the documentation. The DUT is a cover sheet that lists the pages being changed by number and references PR's and other information related to the change. DUT's are used to accumulate changes before their submittal via ECP/SCN or formal publication.

h. Request for Deviation/Waiver

A request for Deviation or Waiver is used by a contractor to request and document temporary departures from configuration identification requirements when permanent changes are not acceptable. A deviation is a written authorization granted prior to product development to permit a contractor to depart from a particular performance or design requirement for a specified product or period of time. A waiver is a written authorization to deliver a configuration item that has been found after development to depart from specified require-

h. Cont.

ments but that nevertheless is considered suitable for use or rework. A proposed design change (ECP) sometimes is converted to a deviation or waiver, or vice versa.

Change Control Process

Class I Change Control

Requests for Class I changes may be initiated by the contractor, AFAL, or other contractors. Class I changes require an ECP and SCN (with draft change pages) to be submitted for approval to the IDAMST CCB. If the change is approved, AFAL issues a Contract Change Notice (CCN) updating the contract, and the contractor releases the SCN with attached specification change pages. The Class I change processing cycle is included in Figure 14.

Class I design changes result from a new or modified requirement outside the scope of the existing contract.

For all Class I design change requests initiated by the contractor, a Preliminary ECP (PECP) accompanied by an SCN with draft change pages to the development specification and an estimate of schedule and cost must be submitted to the IDAMST CCB for approval.

If agreement is reached on the requested Class I design change, a CCBD is transmitted to the procuring organization to issue a CCN approving the contents of the PECP, the SCN, and the schedule and cost estimates.

The detailed design change is prepared in the form of an update to the preliminary product specification. This material supports the CDR, and its further update with the product after development testing. After the Physical Configuration Audit, the final ECP is prepared with the SCN for the product specification.

Implementation of every Class I design change follows a independent baseline development cycle until it can be merged with the current development system.

Class I compatibility changes are changes to correct a deficiency that prevents the product from performing within contractual requirements. They do not increase contract cost.

Procedures for preparing and reviewing Class I compatibility changes are similar to those required for Class I design changes, but simpler. AFAL approval of the PR is necessary before corrective action is initiated. Occasionally AFAL may decide it is better to tolerate the problem addressed by the compatibility change than to risk possible harmful consequences to system interfaces or schedules.

After AFAL approval, a proposed Class I compatibility change can be implemented, tested, and audited. An ECP and SCN are then prepared and submitted to AFAL.

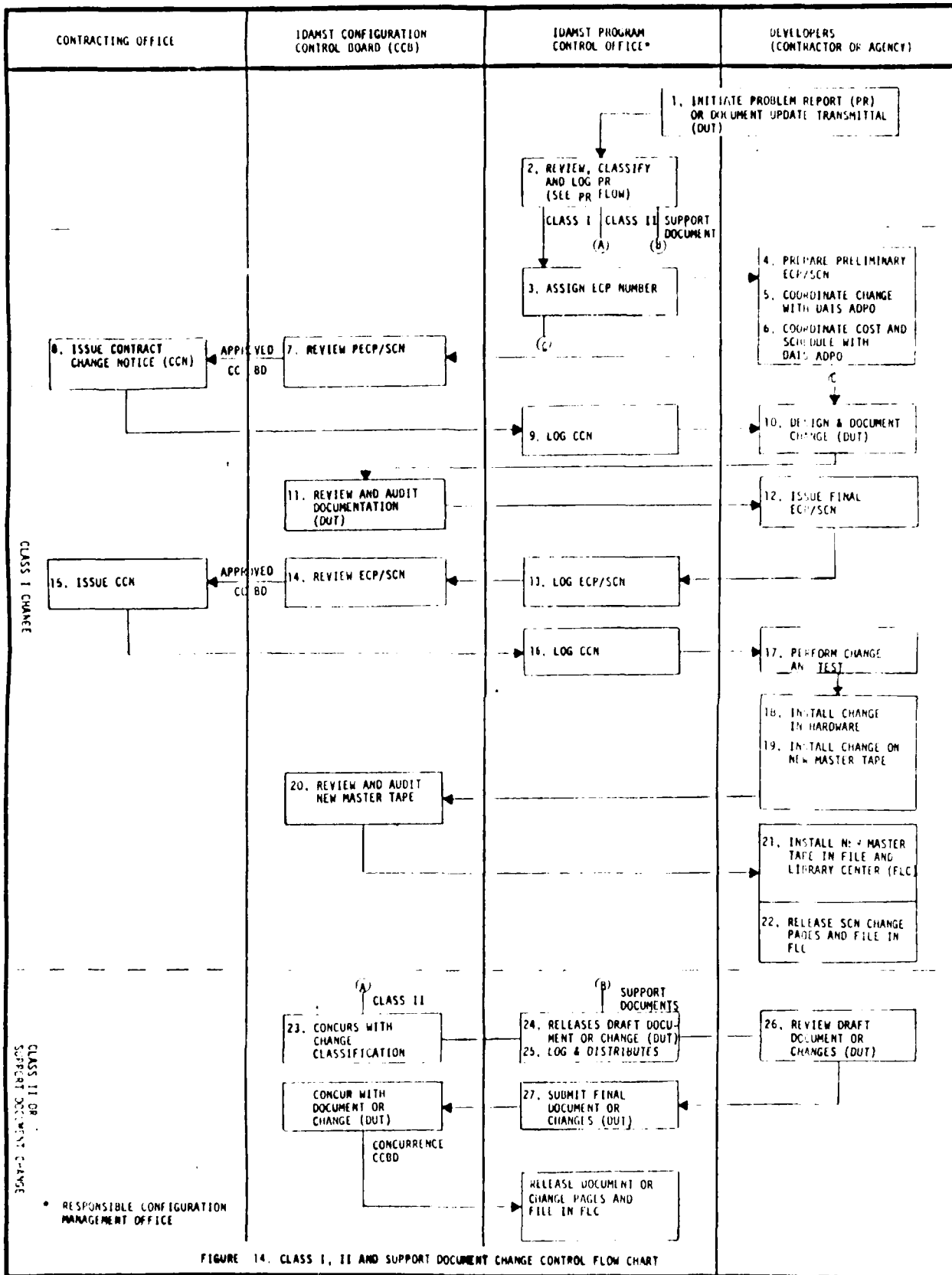


FIGURE 14. CLASS I, II AND SUPPORT DOCUMENT CHANGE CONTROL FLOW CHART

MCDONNELL DOUGLAS CORPORATION

Class II Change Control

Requests for Class II changes may be initiated by the contractor AFAL, or other contractors. Class II changes do not require ECP's or IDAMST CCB approval, but do require IDAMST CCB concurrence. The Class II change processing cycle is included in Figure 14. Class II changes are released with attached SCN(s) or DUT(s), as applicable. Release of the Class II change is authorization to implement the change.

Changes to support documents (except the data base document) are Class II Changes unless they also impact the CI design reflected in a baseline specification. In this case, the Class II change request against the support document should be cancelled and replaced with a Class I change request against the affected baseline specification and/or CI, with the impact on the support document stated therein. Changes to the data base document can be either Class I or Class II.

Release Function

Release (or turnover) of a software product or document is defined as the transfer of the item from the originating organization or contractor, after IDAMST CCB approval or concurrence, to the File and Library Center (FLC) for storage and distribution.

Official release of documentation typically occurs when the documentation has been approved at SDR, PDR, CDR, internal delivery to the IDAMST integration and test team, product baseline, and demonstration baseline. Approved updates of products or documents are considered "re-releases."

Interface Control

"Interface" is defined as the functional and physical characteristics between two or more software programs or pieces of hardware that are not directly controlled by a single contractor or governmental agency, but nevertheless must be assembled and/or function together.

In the IDAMST Program, the interfaces between hardware/hardware, software/hardware, and software/software shall be defined in the Specification Tree and the responsibilities defined in the Work Breakdown Structure (WBS). These interfaces shall be documented in the interface specification or Interface Control Documents (ICD) to a level of detail sufficient to form the basis for design at the allocated baseline. Changes to this documentation must be coordinated with the ICWG and approved by the IDAMST CCB.

Interface Control Working Group (ICWG)

The ICWG is the official communications link among contractors, the AFAL, and other agencies to resolve interface problems, exchange new interface information, document interface change agreements, and coordinate Class I change requests affecting interfaces. The ICWG consists of at least one member from each of the contractors and agencies participating in the system development. These members must have approval authority to commit their organization to a technical sign-off. The system engineer is usually the contractor representative.

The primary ICWG member shall be: (TBD)

The ICWG chairman will determine:

- a. If an interface control requirement need be called to the attention of the ICWG for further investigation.
- b. Which government agencies and contractors will participate in the ICWG investigation.
- c. Action items and assign these items to ICWG members for investigation.

The ICWG secretariat responsibilities shall be:

- a. To schedule and document ICWG meetings, actions and updates.
- b. To issue a meeting agenda five working days before each scheduled meeting.
- c. To issue meeting minutes no later than five working days after each scheduled meeting.
- d. To notify the ICWG chairman of each interface control requirement for determination of further action by the ICWG.

Interface Control Processing

Interface changes are initiated via a Problem Report (PR) or Document Update Transmittal (DUT). The originator prepares and sends a PR or DUT defining the specific interface change. The PR or DUT will be used by the ICWG to describe the disposition of the interface change and to alert holders of interface specifications or interface control drawings interface changes.

At the beginning of each ICWG session, all open interface PR's/DUT's are reviewed. The ICWG chairman makes the final approval or disapproval decision on the interface change. A detailed procedure for interface change control through an ICWG is described in Figure 15.

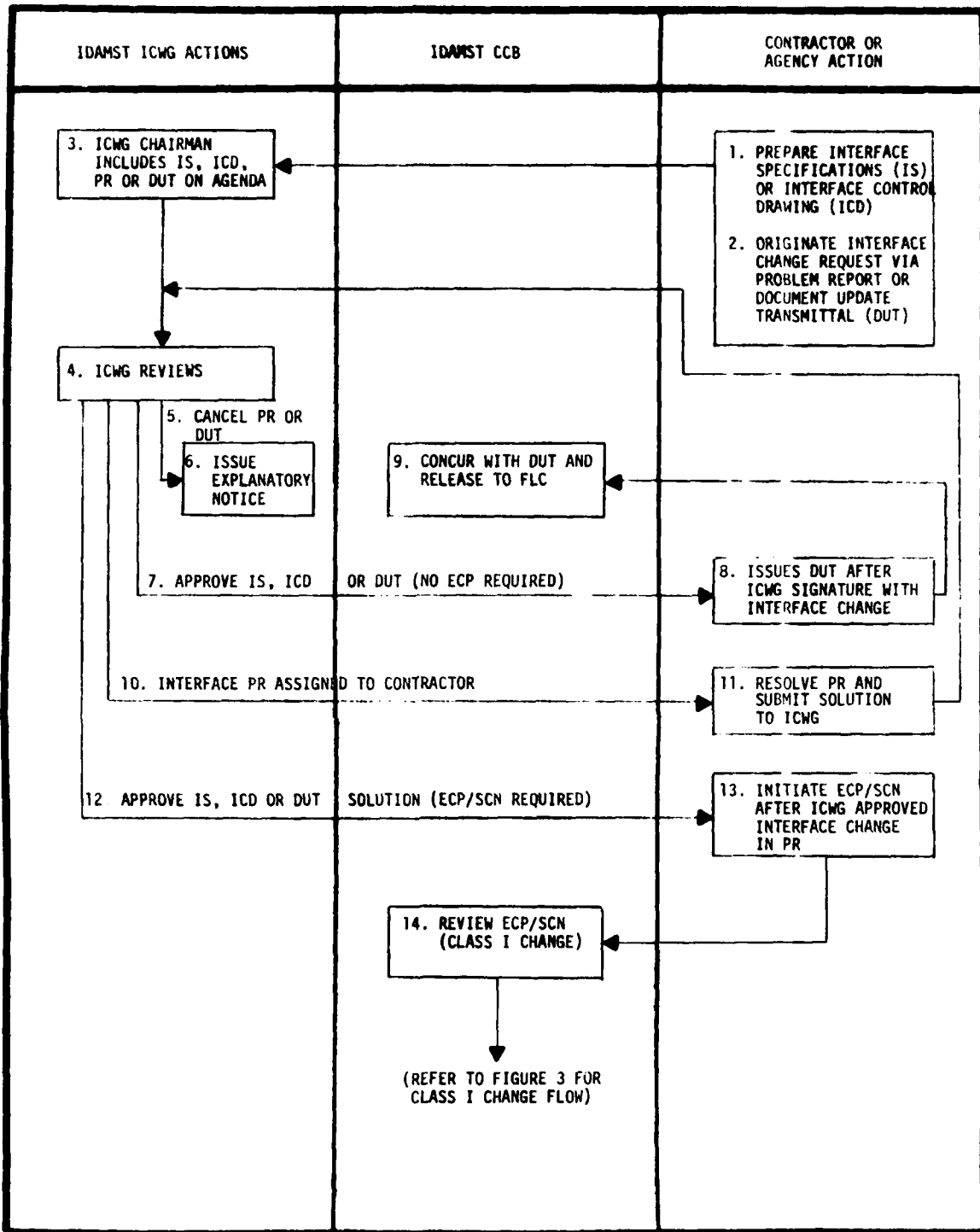


FIGURE 15. ICWG INTERFACE CHANGE CONTROL FLOW CHART

Software Configuration Verification with Checksum

An essential tool for ensuring that an identification software product has not undergone an unauthorized change is the CHECKSUM program (or an equivalent program for computers and operating systems not compatible with CHECKSUM).

When a module is released into accountability under configuration management, the CHECKSUM program is used to calculate a unique signature checksum for the module as derived from the object or load module. A specific checksum is maintained for each version of the module and is identified with the module name, version number, checksum value, and calendar date of release.

Additional checksums are calculated to identify the configuration of the software at the CPCI level as well. These checksums are derived from the disk or tape library, which contains the version of each CPCI. The checksums provide accountability for the ordering of certified modules and data blocks in the linkage edit process. A configuration baseline of checksums should be maintained at both the module and CPCI levels for each release of the software elements. These baselines allow traceability of program modifications for the lifetime of the project and provide a means of identifying old versions of the software for retest purposes.

In addition to its use for configuration verification, CHECKSUM can be used to monitor the software for inadvertent modifications. It can be used to interrogate the libraries residing on the disks and tapes and to produce reports identifying new entries added (modules or data sets), entries deleted, and old entries whose checksums do not match the certified values maintained by configuration management. Validation testing is conducted only with certified copies of the master file. The actual controlled configuration (CHECKSUM) of the software being tested should be noted for all tests.

Version Control of CPCI Elements

More than one version of a software element (module, task) for a deliverable CPCI may exist and be controlled at the same time under different baselines. Each version of an element under multiple baseline control is called a product line. Multiple baselines can occur under the following circumstances:

- a. Follow-on development for subsequent IDAMST Mission requirements.
- b. A product developed in two phases, each independently, such as initial operating capability and full operating capability.
- c. An addition to a product currently in development. For example, an ECP is issued by AFAL to expand capability of a product currently at the product baseline, with the new version to be started at the allocated baseline.

A major problem in multiple baseline control is the coordination of changes among the product lines. For example, product line A, currently in system testing, has system, allocated, and product baselines and is subject to Class I and II change control; while product line B, in validation testing by the contractor, has only system and allocated baselines and is subject only to internal change control. A change of small consequence to product line B might have severe implications for product line A. Therefore, all product lines must be considered whenever a change is incorporated in any of them. To ensure that such consideration has been given, all resolved PR's should be logged against all affected product lines.

Configuration control of each version of a CPCI shall be accompanied by a Version Description Document (VDD).

Problem Reporting and Tracking System

A separate problem reporting and tracking system will be utilized to report, analyze, and track milestones, tasks, and events or technical problems. These problems include, for example, schedule incompatibilities as a result of schedule and milestone analysis, interface incompatibilities as a result of design reviews, test anomalies as a result of test data analysis and component shortage as a result of late delivery or excessive rejection, et al. These are problems of enough significance to require specific emphasis, control, and follow-up to ensure appropriate completion of the action. Problem Reports (PR's) will be listed on the Problem Report Log to provide a proper tracking of action required and performed to resolve the problem at the IDAMST status meetings.

As problem reports are identified, they will be evaluated and classified into two categories. The urgency codes to be used are as follows:

a. Red Flag Problem

A problem requiring expedited action either to solve or to identify a specific solution plan. Time available: 24 hours

b. Normal Problem

A problem requiring action to solve; or to identify a specific solution plan; or investigation to determine the magnitude and urgency of the situation as assigned by appropriate manager.

6.0 CONFIGURATION STATUS ACCOUNTING

Configuration status accounting is the recording and reporting of data concerning a configuration item's identification, proposed changes to its configuration, and the implementation status of approved changes.

The purpose of configuration status accounting is to report the identification status of the evolving hardware and software configuration at periodic intervals throughout the IDAMST Program.

The Major elements of the IDAMST configuration status accounting system are:

- a. Records that contain (1) traceability of all problems to their corrective action, (2) current status of product and documentation, and (3) status of proposed and approved changes.
- b. Reports that provide IDAMST AFAL with essential data on configuration identification and control.

Status accounting records include a log for each configuration control form, with sufficient cross-referencing between logs to facilitate tracing and preparation of various kinds of reports.

The recording elements of status accounting are a series of logs on which change information is recorded daily. Some logs are used for product information, others for documentation information.

Product Logs

The logs required to maintain and track the various elements of a configuration item are listed below. Cross-referencing between logs is necessary for reporting purposes. Examples of these logs are shown in Appendix A.

Software Product Log

The product log is used to list and describe existing tapes, decks, and listings of released CPCI's:

- a. Tape number with prefix identifying the tape (e.g., binary, source, print) and suffix specifying the modification level of the tape with data base identification number.
- b. Routines on the tape with their IDs and modification level.
- c. Listings and tape controlled and numbered by the program name, ID, and modification level.
- d. Originator name and date.
- e. Tape library number.

- f. Distribution information such as copy numbers and location or person checked out with the listings, tapes, etc.
- g. Comments.

Software Module Log

The module log is a running history of each module's development. All activity concerning each module is recorded on separate sheets:

- a. Routine name and ID.
- b. Modification (mod) level with associated PR/SCN/DUT/DBCR numbers that changed each mod.
- c. Test case associated with PR's.
- d. Tape number and tape library number.
- e. Product specification title and number in which the module is published.
- f. DUT and ECP/SCN numbers involving this routine that are outstanding or incorporated.

Data Base Change Request (DBCR) Log

The DBCR Log is used to list and describe all DBCR's:

- a. DBCR number and date.
- b. DBCR originator's name.
- c. Data block identifier.
- d. Data base identification.
- e. Listing of inputs.
- f. Identification and date of data base that incorporates the change.
- g. Comments.

Documentation Logs

The documentation logs are described below and illustrated in Appendix A.

Engineering Change Proposal (ECP) Log

The ECP Log is used to list ECP's and related SCN's and CCN's:

- a. ECP number and SCN number.

- b. Originator's name
- c. Log and submittal dates
- d. Baseline in which ECP was originated
- e. Type of ECP (design or compatibility)
- f. Specification title and number
- g. Technical and cost (if applicable) CCN numbers and date
- h. Data and revision letter of change pages issued to specification

Documentation/Specification Change Notice (SCN) Log

The Documentation/SCN Log is used to list specifications or documents being changed, together with related SCN's, ECP's, and DUT's:

- a. Specification titles, date, and number with all revisions.
- b. Software modules described in the document (CPCI's only).
- c. Outstanding and incorporated DUT's, ECP's, and SCN's against each document.
- d. All support documentation by title, date, and number. Also outstanding and incorporated DUT's, and SCN's.

Problem Report (PR) Log

The PR log is used to list and describe all PR's submitted by AFAL, contractors, or other government agencies:

- a. PR number
- b. Originator
- c. Classification
- d. Problem Description
- e. Impact if not resolved
- f. Proposed action
- g. Accepted solution
- h. Due date

Document Update Transmittal (DUT) Log

The DUT Log is used to list and describe all DUT's:

- a. DUT Number and date
- b. Document title and number being proposed for modification
- c. PR numbers associated with the change

Document Catalog

The Document Catalog provides a complete list of IDAMST generated documents. The Document Catalog is used to identify available literature and as a historical record of delivered documentation.

The catalog contains the following:

- a. Specification or document titles, numbers, dates, and all revisions, addenda, or change notices published.
- b. A list of all modules documented in each CPCI, with ID and current modification level.
- c. All technical support documents (e.g., users manual, data base documents, test plans, etc.)

The Version Description Document accompanies the release of each version of a software configuration (CPCI) and each release of an interim change (i.e., changes that occur between CPCI versions). It identifies the items delivered and records additional pertinent data relating to status and use of the CPCI or change.

Format of the VDD is prepared by the developing contractor or agency and approved by the IDAMST CCB, subject to the incorporation of the following minimum elements of information.

The title page includes the following:

- a. CPCI nomenclature
- b. System title
- c. CPCI number
- d. CPCI specification number
- e. The new version number for each release of a new CPCI version.

- f. For release of a change (interim Version Description Document), the ECP number, SCN/change package designator, and a reference to the current CPCI version for which it is to be attached (e.g., "Version Description Document #1A", with the "1" representing the first version and "A" representing the first interim change).

Content of the VDD should be as follows:

- a. Inventory of Materials Released

List all items (tapes, cards, disks) which are covered by CPCI number and version/number/CPCI number, version number, and change package designator in the case where a new version is not necessary. All utility and/or support computer program release documents not part of the released items but required to operate, load, or regenerate the released CPCI are identified.

- b. Inventory of CPCI Contents

Identify all computer programs and data that are being released, either by reference to appropriate specifications and manuals and/or by listing.

- c. Editorial Changes Installed

A listing of new editorial changes to the computer programs and data base incorporated since the previous version or change, with a cross reference to appropriate specifications. Indicate for each entry in this list the ECP number and date and the related SCN/change package designator number and date. This information does not apply to an initial release.

- d. Class I Changes Installed

Identification of all ECP's reflected in the version and/or change content. Indicate for each entry in this list the ECP number and date and the related SCN/change package designator number and date. For version deliveries subsequent to the initial version, this section lists new ECP's incorporated since the previous release.

- e. Adaptation Data

For the release of a new CPCI version, identify (by reference to appropriate specifications and/or listings) all unique-to-site data in the items being released. For a CPCI version subsequent to the initial version, this section contains the necessary information on changes which have been made to the adaptation data. For the release of a change to a CPCI prior to the issuance of a new version (interim version) this section identifies all changes to the adaptation data as a result of the change.

f. Interface Compatibility

For the release of a new CPCI version, indicate other systems and/or CI's affected by the changes incorporated in this new release. For the release of a change (between versions) to a CPCI, indicate other systems and/or CI's affected by the change.

g. Installation Instructions

Describe (either directly or by reference) the method used to install and check out the delivered CPCI version or change.

h. Possible Problems and Known Errors

Aspects of the change which should be further tested are identified. Any possible problems or known errors are described and any steps being taken to resolve the problems or correct the errors are stated.

7.0 SUBCONTRACTOR CONTROL

Subcontractor control involves several functional areas, one of which is configuration management. This section describes the controls imposed upon the subcontractor to assure that the configuration management requirements of the prime contract will be met. In establishing these controls particular care must be taken to avoid over-control of the subcontractor, resulting in unnecessary cost and restrictions which interfere with the orderly development of the product, or under-control, resulting in inadequate visibility of subcontractor actions and a lack of required documentation. The basis for subcontractor configuration management control is established at the time of subcontractor selection and contract award thru:

- a. Review of the subcontractors configuration management capabilities and procedures.
- b. Inclusion of configuration management documentation requirements in the subcontractors data requirements list (SDRL).
- c. Provision in the statement of work (SOW) for both formal reviews and audits, and informal/in-process reviews.

Following contract award, subcontractor surveillance monitors the performance of the subcontractor relative to contracted configuration management requirements.

SUBCONTRACTOR CONFIGURATION MANAGEMENT REVIEW

Before onsite evaluation, the configuration management response submitted with the subcontractor's proposal must be reviewed to determine the following

- a. That the statements are responsive to the configuration management requirements and data item requirements specified in the RFP.
- b. That confirmation is made relative to the availability of resources to comply with the configuration management requirements specified in the RFP.
- c. That any non-compliance or request for deviation will not impact commitments in the prime contract.

ON-SITE EVALUATION

An on-site evaluation is conducted to verify that the subcontractor has the resources and capability to comply with the RFP configuration management requirements. The on-site evaluation includes the following:

- a. Brief but comprehensive discussions with the subcontractor's CMO manager to verify an adequate level of experience and competency and a thorough understanding of the configuration management requirements.

- b. cursory review of the subcontractor's CMO policies and procedures relative to configuration identification, change control, status accounting, and subcontractor control.
- c. A conducted tour through the subcontractor's configuration management area. This tour should cover.
 - o Review of operations in the document control center relative to document identification, storage, retrieval, distribution, and control of master copies during change incorporation and updating.
 - o Review of operations of the product control center or library relative to product identification, storage, retrieval, distribution, and control of master tapes, disks, etc., during change incorporation and updating.
 - o Review of change control operations relative to change request identification, review, evaluation, approval, distribution, incorporation, and verification of incorporation in documents and products (tapes, disks, etc.)
 - o Review of status accounting operations relative to record keeping, updating of records, and reliability of data in the records.

SUBCONTRACTOR DATA REQUIREMENTS LIST (SDRL)

The RFP contains an SDRL that specifies all data items that the subcontractor must prepare and deliver to the prime. The SDRL also specifies the data item formats, contents, and delivery dates. For configuration management, the SDRL specifies the following data items as applicable:

- a. Configuration Identification Data Items
 - o Development specification
 - o Product specification with flow charts and listings
 - o Interface specification
 - o Data requirements specification
- b. Configuration Control Data Items
 - o Class I change requests for DAC approval prior to implementation
 - o Class II change requests for DAC review after implementation
 - o Change Instructions and Change Notices used for specifying Class I and Class II changes to be made to documents and products
- c. Configuration Status Accounting Data Items
 - o Product log
 - o Routine log
 - o Software Problem Report (SPR) log
 - o Software Modification Record (SMR) log
 - o Data Base Change Request (DBCR) log

- c. Cont.
 - o Engineering Change Proposal (ECP) log
 - o Specification Change Notice (SCN) log
 - o Design Problem Report (DPR) log
 - o Document Update Transmittal (DUT) log
 - o Turnover letter
 - o Version Description Document (VDD)
 - o Product Status Report
 - o Open SPR Report
 - o Document Catalog
 - o CPI Index

All surveillance functions cannot reasonably be performed by any one person or functional group. The surveillance functions must be allocated to the appropriate project group, such as configuration management, product assurance, test management, planning, subcontract administration, etc.

Personnel engaged in subcontractor surveillance perform the following functions as a minimum:

- a. Review subcontractor data items upon delivery to determine compliance with applicable requirements.
- b. Review subcontractor reports relative to cost and schedule status, design reviews, testing, problems, discrepancies, etc., to determine required follow-up action.
- c. Review subcontractor ECPs to determine completeness prior to action of the project CCB.
- d. Review subcontractor Class II change requests to determine correct classifications.
- e. When requested by the subcontractor, represent the prime contractors point of view as a member of the subcontractor CCB during review of Class I changes involving significant cost, schedule, and/or design impact.
- f. Act as a member of the project CCB during review of subcontractor ECPs.
- g. Attend scheduled formal reviews and audits at the subcontractor's facility to determine design status and compliance of the CPI element with applicable design documentation.
- h. Attend scheduled formal testing of the CPI element to determine compliance with applicable test procedures and functional requirements.
- i. Perform periodic audits of the subcontractor's configuration management organization and procedures to determine compliance with contractual requirements.

j. Initiate and follow-up all required corrective action relative to noted problems and discrepancies.

k. Initiate and maintain applicable records and files.

MANAGEMENT INTEGRATION

When geographical and contracting considerations permit, it is highly desirable to integrate the prime and subcontractor management functions. For example, in the area of configuration management this could be accomplished by assigning the subcontractors configuration manager to the software configuration management function under the prime software project manager. The software change control board (SCCB) would be chaired by the software configuration manager who would also be a member of the project CCB and ICWG. Members of the SCCB would be drawn from the subcontractors management functions with a representative from the prime contractor, most likely the software system engineering manager.

8.0 CONFIGURATION AUDITS AND REVIEWS

A series of configuration reviews and audits provide verification that the performance achieved by the product is the performance required by the development specification, and that the configuration of the product is accurately specified in the product specification. These reviews and audits must be scheduled at meaningful points (Figure 4.1) during the development of a configuration item to permit assessment of progress and to establish new baseline configuration identifications for the product.

The Functional Configuration Audit (FCA) and Physical Configuration Audit (PCA) are primarily system engineering oriented, with less emphasis on CM aspects. For the Preliminary Design Review (PDR) and Critical Design Review (CDR), the CM objective is to verify that required changes are incorporated in the applicable baseline specifications and that the updated baseline specifications accurately specify the approved design requirements for the next phase of CI development.

The contractor or agency project manager or his designee is co-chairman for all reviews and audits. The other co-chairman is an AFAL IDAMST representative.

The review material is distributed to AFAL IDAMST project personnel at least ten (10) working days prior to the meeting. The IDAMST project personnel and contractor personnel shall examine the material thoroughly and identify all deficiencies. (One of the most important factors in the achievement of satisfactory reviews and audits is the depth and completeness of technical reviews.)

Unless otherwise specified in the contract or in an understanding with the IDAMS Chief Engineer, the reviews and audits are conducted at a contractor's or agency's facility. The contractor or agency project manager provides the necessary resources and material to effectively conduct the review or audit. This includes, but is not limited to, the following items to the extent appropriate:

- a. Meeting agenda and plans
- b. Conference room or rooms
- c. Applicable system engineering data, specifications, hardware/software interfaces, schedules, and design and test data.
- d. Special study results
- e. Minutes of the meeting, including action items

After each review or audit, the AFAL co-chairman provides AFAL's position as to the approval or disapproval of the review or audit:

- a. Unqualified Approval

Signifies complete agreement between reviewing agencies and the contractor.

b. Approval with Contingent Action Items

The review is not considered accomplished until action items have been satisfactorily completed as determined by AFAL co-chairman.

c. Approval with Deviations

This action is used as approval when it is recognized that specific CI specifications have not been met. This action is used when it is in the AFAL interest to provide limited approval to assure program schedules. Formal deviations must be submitted by the contractor through the configuration management channels to IDAMST CCB.

d. Disapproval

This action is used when the configuration item is unsatisfactory or generally inadequately identified, or generally doesn't meet the CI specification. A new review must be scheduled to accomplish the action items on that CI.

After completion of the review or audit, the minutes are published and distributed within five (5) working days by the contractor or agency. Notification of any required post-review action items, plus official acknowledgment of approval/disapproval of the review, is provided by AFAL within ten (10) working days after receipt of the minutes. Problems identified at the review or audit shall be submitted as Problem Reports to the IDAMST Program Control Manager by the AFAL co-chairman.

System/Subsystem Requirements Reviews (SRR)

Purpose of SRR

System/subsystem requirements reviews (SSR's) are also called system/system segment requirements reviews. SRR's are in-process reviews usually conducted on concept definition contracts for a new large-scale system. Their purpose is to assure the customer that the project's output is responsive to the accomplishment of the functional analyses and preliminary requirement allocation and to determine initial direction and progress of the system engineering effort and convergence upon a system configuration.

SRR Review Items

The principal items to be reviewed at the SRR are:

- a. Statement of Work
- b. System specification
- c. System engineering studies considering mission and program requirements analyses, preliminary requirements allocation, trade studies, system interface studies, test planning, etc.

System Design Review (SDR)

Purpose of SDR

The SDR evaluates the optimization, traceability, correlation, completeness, and risk of the allocated functional requirements (allocated configuration baseline), including the corresponding test requirements, in fulfilling the system or system segment requirements. The review encompasses the total system requirements items used to produce the system definition (CI development specifications) and should be the final review before submittal of the definition phase products. This review establishes the allocated baseline for individual CI's upon approval of each development specification and initiates contractor configuration activities and procedures.

SDR Review Items

The principal items to be reviewed at the SDR are:

- a. System specification with change pages.
- b. Development specifications for each CI.
- c. Tradeoff studies, analyses, etc.

During the definition phase, the development specifications and test plans are continually updated. At completion of the definition phase, a final SDR may be required, at which time IDAMST CCB approves the above documentation. This documentation then becomes subject to configuration control.

Software Preliminary Design Review (PDR)

Purpose of PDR

The PDR is a formal technical review of the basic design approach for a CPCI. The PDR for each CPCI occurs between SDR and CDR. Only one successful PDR per CPCI is required. A collective PDR for a functionally related group of CPCI's, treating each CPCI individually, may be held when advantageous to the project. The overall technical risks associated with each CPCI are also reviewed on a technical cost and schedule basis.

PDR Review Items

The PDR documentation package should include the following:

- a. Updated system specification
- b. Development specification including approved changes
- c. Draft of CPCI and CPC development test plans
- d. Draft of product specifications
- e. Data base structures and organization
- f. Interface specifications

- g. Development schedule
- h. Standardization considerations
- i. Estimated memory requirements and allocation
- j. Results of studies, analysis, and testing

The PDR is conducted to accomplish or establish the following:

- a. Compatibility of the design approach with the development specification of the CPCI and with other system equipment and facilities or other software programs.
- b. Review draft sections of parts of the preliminary CPCI product specification.
- c. Integrity of design.
- d. Adequacy of development test plan.
- e. Verify that approved system specification and development specification changes are reflected in the draft product specification sections.

The items and tasks to be considered include but are not limited to the following:

- a. Complete computer program functional flow to the level of flow charting that identifies allocation of computer program components to required functions and depicts the sequence of operations within the system functional flow.
- b. Detail storage allocation charts for the CPCI as a whole, describing the manner in which available storage is allocated to individual computer program components (CPCs). Include timing, sequencing requirements, and relevant equipment constraints.
- c. Describe control functions including the executive control and start/recovery features for the computer program system, method of initiating system operation, and features that permit recovery from system malfunction.
- d. Describe structure and organization of the data base to a level that identifies data types and characteristics, structure layout, and allocation of data storage.

The form of the PDR could consist of a briefing with interrogations and discussion thereafter, or a walk-through of documentation, or a combination.

Software Critical Design Review (CDR)

Purpose of CDR

The CDR is a formal review conducted on each module of each CPCI including the flow charts, logic, and algorithms to the higher order language (HOL). The CDR ensures that the detail design solution, as reflected in the draft product specification, satisfies performance requirements established by the development specification. The CDR also establishes system design compatibility with regard to other CPCIs and within the CPCI.

This is not intended to preclude the release of portions of complex CPCIs for coding when this is necessary to maintain schedule or to verify difficult design components. In fact, CDRs for the CPCs of complex CPCIs usually are conducted on an incremental basis.

Representatives of other contractors or subcontractors responsible for design or development of items that interface with CPCIs may participate in the CDR.

CDR Review Items

The CDR documentation package should consist of the following:

- a. Development specification, including approved changes.
- b. CPCI and CPC development and validation test plans/procedures.
- c. Data maps
- d. Memory requirements and allocation.
- e. Draft of operation manual.
- f. Draft of maintenance manual.
- g. Draft of the product specification, excluding listings.
- h. Appropriate support documentation, such as updated timing studies, accuracy studies, results of testing, etc.
- i. Applicable configuration management records related to approved changes to the development specification.

The tasks to be performed are the following as a minimum:

- a. Establish compatibility of design with the development specification.
- b. Establish system compatibility of design and review all interfaces between CPCIs and between CPCs within a CPCI.
- c. Analyze interactions with data base.
- d. Establish design integrity by review of available test and analytical data in the form of logic diagrams, algorithms, storage allocation charts, detailed flow charts, etc.

- e. Review interfaces between CPCI and the applicable hardware CIs to ensure that changes have not affected compatibility.
- f. Review updating changes to the system and development specifications subsequent to the PDR, to determine whether the draft product specification adequately reflects these changes.
- g. Review all available test documentation for currency, technical adequacy, and compatibility with Section 4 of the system and development specification requirements.
- h. Review all test documentation required to support test requirements of Section 4 of development specifications (test procedures in particular) for compatibility, technical adequacy, and completeness.
- i. Ensure that plans are initiated for reallocation of specific CPCI functions to different CPCs, when such reallocation is made necessary by actions occurring prior to or during CDR.

Functional Configuration Audit (FCA)

Purpose of FCA

The FCA verifies the CI's actual performance compliance with the development specification. Test data is reviewed to verify that the item has performed as required by its allocated configuration identification identification. FCAs for CPCIs may be conducted on an incremental basis at the function, module, or routine level. For CPCIs that can be validated only through integrated systems testing, the FCA cannot be completed until such testing has been completed.

FCA Review Items

The documentation package to be provided and made available to the customer during FCA consists of:

- a. Draft product specification
- b. Development specification with changes, if applicable
- c. Test plan, test procedures, and test reports
- d. PDR and CDR minutes
- e. Status accounting records

The procedures and requirements pertinent to the FCA comprise review tasks, audit tasks, other action tasks, and analyses. The review tasks consist of examining and evaluating:

- a. Test procedures and results for compliance with the development specification. Testing must verify that data, procedures, and results are sufficient to ensure CI performance to the development specification Section 3.

- b. Adequacy of analysis or simulations where performance parameters cannot completely be verified by test.
- c. Briefing to AFAL on any requirements stated in the development specification that could not be met and contractor or agency proposed solution.
- d. All approved ECPs to ensure they are incorporated and verified during validation testing.
- e. PDR and CDR minutes to ensure that all findings have been incorporated and completed.
- f. Draft product specification to provide guidance for FCA submittal.
- g. Interface requirements and testing.

The audit tasks consist of examining and comparing:

- a. Validation and acceptance test plans and procedures for comparison against official test data. Results are checked for completeness, accuracy, etc. Deficiencies are documented and completion dates of corrective actions established and documented (FCA minutes).
- b. The validation and acceptance test report to verify that it is accurate and completely describes the validation tests.

Physical Configuration Audit (PCA)

Purpose of PCA

The PCA is a formal examination of the CPCI against its technical documentation and of the configuration management records pertinent to the CPCI in order to establish the product baseline. The PCA cannot be conducted unless the customer has the final draft of the product specification. After successful completion of this audit, all subsequent changes are processed by ECP. This includes version deliveries of CPCI elements and documents prepared to accommodate new or revised capabilities and requirements of the CPCI.

The PCA includes a detailed audit of the product specification; including flow charts, listings, and design narratives.

It also includes a review of the format and completeness of the operators manual, maintenance manual, and any other manuals or handbooks due for acceptance at this time. These latter documents are reviewed and analyzed for acceptance subsequent to PCA after system tests have verified that procedures in the manuals are accurate. Configuration management records related to the CPCI will be reviewed to ensure that all approved changes are incorporated and that unincorporated changes are disposed of.

Formal approval by the customer of the CPCI product specification, via a signed DD Form 250 or equivalent, constitutes the product baseline.

PCA Review Items

The data items to be delivered 20 days to AFAL prior to audit consist of:

- a. PCA date and location
- b. Agenda
- c. List of representatives for the performing organization, including the test manager or equivalent
- d. Identification of configuration items to be accepted
 - o Nomenclature
 - o Specification identification number
 - o CPCI identifiers
- e. A list of all requests for deviations, either incorporated with customer approval or outstanding and awaiting customer approval.

The documentation package to be provided and made available to AFAL for the PCA consists of:

- a. Development specification
- b. Product specification
- c. Test procedures and test report
- d. FCA minutes
- e. Version description document (VDD) if applicable
- f. Draft operation manual
- g. Draft maintenance manual
- h. CPCI tape and listings
- i. Configuration management records as applicable
- j. Prepared DD Form 250 or equivalent

The procedures and requirements pertinent to the PCA comprise review tasks, audit tasks, and other action tasks. The review tasks consist of:

- a. Review product specification for format and completeness.
- b. Review FCA minutes for recorded discrepancies that require action.
- c. Review module descriptions, flow charts, interface requirements, coded program, and testing for accuracy and completeness.

The review tasks should be a sampled crosscheck rather than an exhaustive technique.

The audit task consists of an audit of the contractor or agency engineering release and change control system to be sure that it can properly control the processing and formal release of engineering changes. The release system and associated documentation will have the following minimum needs and capabilities:

- a. Identifying changes and retaining records of superseded configurations formally accepted by the customer.
- b. Identifying all Class I and II engineering changes released for incorporation. These changes are completely released and incorporated before formal acceptance of the CPCI.
- c. Determining the configuration released for each CI and CPCI at the time of formal acceptance.

APPENDIX A
CHANGE CONTROL
FORMS
AND
LOGS

CHANGE CONTROL FORMS

FIGURE

Problem Report (PR)
Data Base Change Request (DBCR)
Engineering Change Proposal (ECP)
Specification Change Notice (SCN)
Configuration Control Board Directive (CCBD)
Contract Change Notice (CCN)
Document Update Transmittal (DUT)
FLC Withdrawal Request (WR)

TBD

LOGS*

Problem Report Log
Software Product Log
Software Module Log
Data Base Change Request Log
Hardware Product Log
ECP Log
SCN Log
DUT Log

TBD

CATALOG*

Document Catalog

TBD