

ESD-TR-77-254

13
B.S.



AN AIR FORCE GUIDE TO COMPUTER
PROGRAM CONFIGURATION MANAGEMENT

System Development Corporation
2500 Colorado Avenue
Santa Monica, CA 90406

August 1977

DDC
RECEIVED
DEC 6 1977
F

Approved for Public Release;
Distribution Unlimited.

Prepared for

DEPUTY FOR COMMAND AND MANAGEMENT SYSTEMS
ELECTRONIC SYSTEMS DIVISION
HANSOM AIR FORCE BASE, MA 01731

AD A 0 4 7 3 0 8

FILE COPY

LEGAL NOTICE

When U.S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

OTHER NOTICES

Do not return this copy. Retain or destroy.

This technical report has been reviewed and is approved for publication.

William J. White

WILLIAM J. WHITE, Capt, USAF
Project Engineer

John C. Mott-Smith

JOHN C. MOTT-SMITH, GS-13
Project Leader

John T. Holland

JOHN T. HOLLAND, Lt Col, USAF
Techniques Engineering Division

FOR THE COMMANDER

Frank J. Emma

FRANK J. EMMA, Colonel, USAF
Director, Computer Systems Engineering

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD TR-77-254	2. GOVT ACCESSION NO.	3. REPORTING ORIGIN NUMBER (9) Technical rept.
4. TITLE (and Subtitle) AN AIR FORCE GUIDE TO COMPUTER PROGRAM CONFIGURATION MANAGEMENT,		5. TYPE OF REPORT & PERIOD COVERED
7. AUTHOR(s) Lloyd V. Searle		6. PERFORMING ORG. REPORT NUMBER
8. CONTRACT OR GRANT NUMBER(s) FI9628-76-C-0236		9. PERFORMING ORGANIZATION NAME AND ADDRESS System Development Corporation 2500 Colorado Avenue Santa Monica, CA 90406
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS		11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Command and Management Systems Electronic Systems Division Hanscom AFB, MA 01731
12. REPORT DATE Aug 87		13. NUMBER OF PAGES 147p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A		16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release; Distribution Unlimited.
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Computer Program Configuration Management Software Management Controls Computer Program Specifications Software Documentation Software Acquisition Management Software Configuration Management		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This document contains guidance to assist Air Force activities in applying practices and procedures of configuration management to computer programs. It is written as a combined instructional and reference document which identifies, interrelates, and explains the requirements of current Air Force and DoD configuration management standards as they apply during the acquisition of a major military system. The guide is organized into a series of sections covering: background information; principles involved in the selection and identification of computer program configuration items; characteristics and functions		

20. Con't.

of specifications; practices of configuration and interface control; significant aspects of computer program document maintenance and status reporting; and considerations involved in control during the system test period.

ACCESSION for		
NTIS	1	on
DDC	B	ET
UNANNOUNCED		
JUSTIFICATION		
BY		
DISTRIBUTION/AVAILABILITY NOTES		
Dist.		DATE
A		

PREFACE

This guidebook is one of a series being developed under the sponsorship of the Electronic Systems Division, Air Force Systems Command, directed for ESD by the Directorate of Computer Systems Engineering (ESD/MCI).

The purpose of the series as a whole is to supplement other measures being taken by the Air Force and Office of the Secretary of Defense to improve the management of computer resources in defense system programs. Within the Air Force, emphasis is placed on providing information in a form to support the effective implementation of policies set forth in the 800-series Air Force regulations. In this series sponsored by ESD, further emphasis is devoted specifically to software elements of programs to acquire the command, control, and communications (C³) class of systems.

Configuration management is one of several topics for which guidebooks are being planned or prepared under contract with the MILRE Corporation and System Development Corporation. It is contemplated that the Software Acquisition Management (SAM) series as a whole, when completed, will cover the following topics:*

- Regulations, Specifications and Standards (AD-A016401)
- Contracting for Software Acquisition (AD-A020444)
- Monitoring and Reporting Software Development Status (AD-A016488)
- Statement of Work Preparation (AD-A035924)
- Reviews and Audits
- Computer Program Configuration Management
- Computer Program Development Specification (Requirements Specification)
- Software Documentation Requirements (AD-A027051)
- Verification

*National Technical Information Service accession numbers shown in parentheses identify topics for which guidebooks have already been published. For full titles, authors, and other identification of these guidebooks, see Section 8, Bibliography.

- Validation and Certification
- Overview of the SAM Guidebook Series
- Software Maintenance
- Software Quality Assurance
- Software Cost Estimation and Measurement
- Software Development and Maintenance Facilities (AD-A038234)
- Life Cycle Events (AD-A037115)

As those titles may imply, the concern of the guidebooks is not with computer programming standards or guidance of a technical nature. Rather, it is with how to apply Air Force policy and practices for managing the acquisition of military systems to the software-related elements of those systems.

Thus, the focus is on management as opposed to technical guidance, and on management in the context of Air Force systems as opposed to generalized management of software. At the same time, it is fundamental to the Air Force systems approach that the management techniques must be formulated and applied in a manner which takes adequate account of the technical considerations associated with each major class of system element--whether hardware, software, facilities, data, or people.

This guidebook observes those and other general guidelines established by ESD sponsors for the series as a whole, pertaining to such factors as content, level, and intended audience. The guidance is based, throughout, on current Air Force and Joint Services regulations, specifications, and standards for configuration management as they apply to computer programs. To the best of the author's ability, it also reflects problems, successes, and lessons learned through the actual uses of those documents in a substantial number of electronic system programs over the past decade.

TABLE OF CONTENTS

	<u>Page</u>
PREFACE	1
LIST OF FIGURES	6
LIST OF TABLES	6
 SECTION 1. INTRODUCTION.	 7
1.1 Purpose.	7
1.2 Scope and Organization	7
1.3 Configuration Management: One, Limited Discipline	8
1.4 General Concepts	10
1.5 Source Regulations, Specifications, and Standards. . . .	14
1.6 Special Terms.	16
 SECTION 2. SELECTION OF COMPUTER PROGRAM CONFIGURATION ITEMS	 21
2.1 Requirements and Criteria.	21
2.1.1 Requirements.	22
2.1.2 Selection Criteria.	23
2.2 Pitfalls	24
2.3 The Type Classification.	26
2.4 Identification Numbers and Markings.	26
 SECTION 3. SPECIFICATIONS	 29
3.1 The Specification Tree/CI List	30
3.1.1 System/System Segment Specification Level	31
3.1.2 CI Specification Level.	31
3.1.3 Critical Item Specification Level	32
3.1.4 Specification Tree vs. Generation Breakdown	32
3.2 System Specification	34
3.2.1 Content	34
3.2.2 Development and Control	35
3.3 Computer Program Development Specification	36
3.3.1 Content	37
3.3.2 Development	38
3.3.3 Approval and Functions.	40
3.4 Computer Program Product Specification	41
3.4.1 Content and Development	42
3.4.2 Approval and Control.	44
3.5 Other Specifications	46
3.5.1 Form 3 Specifications	46
3.5.2 Inventory Items	47
3.5.3 Addendum Specification.	48

CONTENTS (cont'd)

	<u>Page</u>
3.6 Summary of Specification Roles, Hardware vs. Software. .	48
SECTION 4. CONFIGURATION CONTROL	53
4.1 Organizational Factors	54
4.2 Change Classification.	57
4.3 Class I Change Processing.	58
4.3.1 Two-Step Processing	59
4.3.2 Preparation of ECPs	68
4.4 Class II Changes	69
4.5 Related Controls	72
4.5.1 Baseline Management as a Technical Tool	72
4.5.2 Engineering Release Systems	74
4.6 Interface Control.	75
4.6.1 General Concepts and Responsibilities	76
4.6.2 Documentation of Software Interfaces.	77
SECTION 5. DOCUMENT MAINTENANCE AND STATUS REPORTING	81
5.1 Document Identification and Maintenance.	82
5.1.1 Document Issues	84
5.1.2 Document and Page Identification.	85
5.1.3 Front-Matter Pages.	86
5.2 Configuration Index.	90
5.2.1 Organization and Timing	91
5.2.2 Preparation of Sections	92
5.3 Change Status Report	96
5.3.1 Section I	97
5.3.2 Section II.	99
5.4 Version Description Document (VDD)	100
5.4.1 Definitions and General Policy.	100
5.4.2 Numbering Versions and VDDs	101
5.4.3 Preparation and Content	101
SECTION 6. CONTROL DURING SYSTEM DT&E.	105
6.1 Assumed Initial Conditions	105
6.2 On-Site Control and Support.	107
6.3 Summary Characteristics.	108

CONTENTS (cont'd)

	<u>Page</u>
SECTION 7. NOTES	111
7.1 Computer Programs as Data vs. CIs.	111
7.2 "Software" and the DoD Directive	113
7.3 System Segments.	117
7.4 Problems with Chapter 2 of AFR 800-14.	119
7.5 Configuration Index - Questions.	121
SECTION 8. BIBLIOGRAPHY.	127
8.1 Regulations, Specifications, and Standards.	127
8.2 Technical Reports.	128
SECTION 9. TERMS AND ABBREVIATIONS	129
9.1 Terms.	129
9.2 Abbreviations.	138

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1-1. Configuration Management - Responsibilities and Limitations . . .	10
1-2. Sequence and Structure of Documents Involved in Configuration Management.	13
3-1. The Specification Tree.	30
3-2. Development of the System Specification	35
3-3. Development of the CPCI Part I Specification.	39
3-4. Development of the CPCI Part II Specification	44
3-5. Hardware vs. Software - Life-Cycle Factors Significant to Configuration Management.	50
4-1. CCB and CMO Relationships to a Program Office	56
4-2. Illustrative Organization for a Software Contractor	56
4-3. Synoptic Diagram of Two-Step Processing	62
4-4. The Computer Program ECP Form	64
4-5. Initiation and Preparation of ECPs Proposing Within-Scope Expansions of the CPCI Part I Specifications.	67
4-6. Initiation and Preparation of a Class I Change Proposal Involving Subsequent Implementation Effort.	68
4-7. Sample Contractor's Form for the Class II Change Report (CR). . .	71
5-1. Sample Front-Matter Pages for Maintainable Documents.	89
5-2. Configuration Index: Sample Table of Contents.	93
5-3. Configuration Index: Sample Section A.	93
5-4. Configuration Index: Sample CDRL Backup Instructions	94
5-5. Configuration Index: Sample Section I.	94
5-6. Configuration Index: Sample Section III.	95
5-7. Configuration Index: Sample Section VI	95
5-8. Change Status Report: Sample Section I	98
5-9. Change Status Report: Summary of Section II Content.	98
5-10. Version Description Document: Summary of Contents.	102
6-1. Relationships Among Activities During System DT&E	106

LIST OF TABLES

<u>Table</u>	
1-1. Significant Source Documents Referenced in the Text	15
2-1. Sources of Requirements for Numbers and Markings.	27
3-1. MIL-STD-490 Appendices for Specification Types and Subtypes . . .	29
5-1. Summary of Identification Data Required for CPCI Specifications .	85

SECTION 1. INTRODUCTION

1.1 PURPOSE

This guidebook is addressed to personnel of Air Force Systems Command program offices who are responsible for managing software-related portions of system programs conducted in accordance with policies of the 800-series Air Force regulations. Within that framework, its purpose is to provide basic instructional and reference materials which will support the effective application of requirements for computer program configuration management.

1.2 SCOPE AND ORGANIZATION

This first section presents information of a background and introductory nature, reviewing general concepts, principles, special terms, and the status of Air Force/DoD configuration management standards. The remainder of the guidebook consists of sections covering the topics summarized briefly below:

Section 2 discusses the requirements and criteria for selecting assemblies of computer program code to be identified as computer program configuration items (CPCIs), and includes a subsection summarizing the sources and coverage of standards for identification numbers and markings.

Section 3 is devoted to specifications. It addresses: specification types and forms; the specification tree; the system specification; computer program development and product specifications; other types/forms of specifications applicable to computer programs; and comparisons between software and hardware with respect to the roles of their specifications in the system acquisition cycle.

Section 4 covers requirements and procedures for processing changes to approved specifications. It identifies organizational factors, explains change classification, describes standard forms, and discusses procedures involved in the preparation and processing of change proposals. It includes a subsection dealing with concepts of interface control and the documentation of interfaces involving software.

Section 5 is devoted to requirements and practices of document identification and maintenance which are significant to configuration management functions, and to formal reports/records of status for documents, change proposals, and CPCIs.

Section 6 addresses factors which come into play following completion of development and initiation of the CPCI operations at a field location. Using a sample system DT&E situation for illustration, it identifies the nature of questions to be anticipated and shows how centralized controls and procedures described in preceding sections relate to that expanded framework.

Section 7 contains notes written in response to questions raised by reviewers of a draft version of this guidebook, pertaining to a few of the topics covered in preceding sections.

The bibliography lists references cited in the text, other guidebooks published to date, and a few other documents consulted by the author during preparation of this guidebook.

The glossary identifies abbreviations used in the text and explains standard Air Force/DoD terms as they apply to configuration management of computer programs.

Thus, with respect to the familiar, major subtopics of configuration management: configuration *identification* is covered in Sections 2 and 3; configuration *control* is covered in Section 4; and the software counterparts of configuration *status accounting* are covered in Section 5. Configuration *audits* are not specifically mentioned in the above summary for the reason that those are assigned as major topics to be covered separately in the Reviews and Audits guidebook. However, selected aspects of both audits and technical reviews are dealt with in the text as necessary to explain their interrelationships with the other configuration management topics under discussion.

1.3 CONFIGURATION MANAGEMENT: ONE, LIMITED DISCIPLINE

Acquisition management is accomplished in AFSC program offices (POs) by a complex of interrelated, but separately identified, management disciplines. The disciplines represent separate areas of management responsibility which correspond, largely, with individual career specialties of PO personnel. They are also the basis for the typical PO organization and for major topics addressed in the various acquisition management regulations, specifications, and standards. Thus, distinctions among assigned areas of functional responsibility, as summarized briefly below, are generally fundamental to practices and interrelationships dealt with in later sections of this guide.*

*For more complete descriptions of these PO functions, see AFSCP 800-3, which provides a separate chapter on each function.

- Engineering - Management of the technical program, including software and other component engineering specialties, system engineering, and human factors.
- Procurement - Legal responsibilities for purchasing and contracts.
- Program Control - Management of program costs and schedules--i.e., estimating, controlling, tracking, and reporting of budgets, costs, schedules, and related management information.
- Test and Evaluation - Planning and control of the development test and evaluation (DT&E) program; coordination and support of operational test and evaluation (OT&E).
- Deployment - For electronic systems, management of system activation.
- Integrated Logistics Support - Planning and management of provisions for deployment phase support of system operations and maintenance.
- Data Management - Identification and control of contractually deliverable reports and other items of data produced for the program.
- Configuration Management is defined in the current Joint Services regulation, AFR 65-3, as:

"A discipline applying technical and administrative direction and surveillance to (1) identify and document the functional and physical characteristics of a configuration item, (2) control changes to those characteristics, and (3) record and report change processing and implementation status."

The remainder of this guide is devoted to further amplifying the scope and practices of configuration management as those apply to computer programs. However, configuration management is essentially a support function which interacts closely with, and depends on the proper conduct of, engineering and the other management disciplines. Hence, those interrelationships must also be taken into account, including the restrictions which they impose on the scope of a configuration manager's responsibilities within the system PO as a whole. Two major sources of limitations to be recognized are represented synoptically in Figure 1-1:

- (a) the configuration manager's most direct concern is with those system elements which are subject to being designated as configuration items; and
- (b) his authority with respect to those elements is further limited to certain special, formalized aspects of their management control.

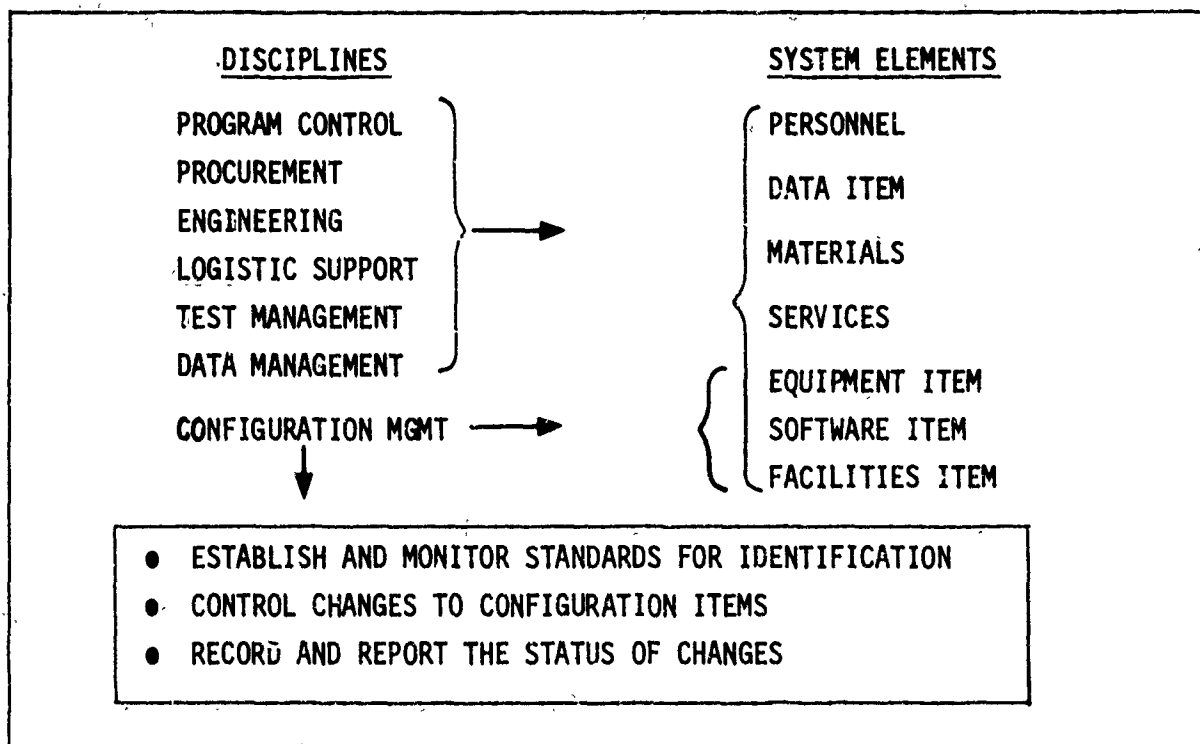


Figure 1-1. Configuration Management - Responsibilities and Limitations

1.4 GENERAL CONCEPTS

The "system elements" identified in the preceding Figure 1-1 are distinguished largely because they involve certain characteristic differences in the proper approach to their acquisition and management. To some degree, they are related to the management disciplines. However, it is significant that all of the disciplines currently represented in system POs were firmly established, with respect to approaches and procedures appropriate to the other system elements, before the prominence of software* in systems became widely recognized.

*Meaning, in this guide, computer programs; see 1.6.5. The term "software" itself was fairly prominent in the 1950s, but it referred then to deliverable items of contractor data, such as handbooks and manuals; that use can still be found in some current regulations.

Problems have been encountered because computer programs represent a relatively new class of system components which are neither quite the same as, nor at the same time totally different from, any of the other elements. A question of long standing is whether a computer program should be acquired as an item of data or as a manufactured product--i.e., in this context, whether to apply procedures of data management, vs. whether to subject it to procedures in such areas as engineering, test, and configuration management. The Air Force decision relating to that question was reached as early as 1963. It is reflected in the existing regulations, specifications, and standards, and has been reaffirmed recently in such documents as AFR 65-3 and AFR 800-14. That decision and some of its logic are outlined very briefly as follows:

- A computer program is intrinsically an item of data--i.e., it is written, recorded, translated, reproduced, etc. in ways that are characteristic of data as opposed to equipment.
- However, its role as an element of the operational system is more like that of equipment; and there are reasons to manage its development through the use of techniques similar to those employed for equipment items, e.g., with respect to specifications, configuration control, interface control, reviews, audits, testing, and the fact that a computer program item is itself, like equipment, the basis for the preparation of operating and support manuals.
- At the same time, established procedures for managing equipment items do not automatically apply; they must be tailored, throughout, to take into account the unique characteristics of computer programs.

Thus, computer programs are presently classified as configuration items, but are also recognized as separate from equipment, for purposes of managing their acquisition as elements of systems.*

In general, a configuration item is an identified facility, equipment item, or computer program item which is specifically designated in a given acquisition as being subject to configuration management. The "configuration" of an item (or system) refers to the totality of its functional and physical properties, which are defined and documented, for practical purposes, in the form of specifications. Thus, specifications serve as the principal documentary instruments for configuration management; and it has become one important function of configuration management, historically, to promote and disseminate uniform standards to govern the types, forms, and levels of description at which specifications are prepared.

*Various differences in procedures and objectives for configuration management of equipment and computer programs are discussed later in the text. A few additional points pertaining to computer programs as data are provided in the note, 7.1.

In a given developmental program, the actual content of each specification results from the mainstream engineering efforts of technical analysis, design, and development. Management control is initiated by establishing a completed specification, formally, as an approved and accountable document at the time of its original issue. Through that action, the configuration described by the specification becomes an explicit point of departure, or baseline configuration, against which changes can be proposed and evaluated.

Activities of formulating and implementing changes to a baseline configuration are also basically technical. Management controls are applied during the change process to assure: that each change proposed is evaluated in relation to all relevant technical, schedule, and cost factors; and that each change approved and implemented is reflected in a corresponding change to the specification, so that the specification continues to define the current approved configuration of the system or item at all times.

As an important part of that general configuration management process, the status of configuration for the system and each item is made known to all participating technical and management activities whose coordinated efforts in developing the system may be affected. This function is accomplished by controlled dissemination to appropriate activities of the specifications, change proposals, updating changes, and periodic status reports.

The term "baseline management" is frequently used to describe the generalized characteristics of that process, which can also be applied usefully in other ways (see 4.5.1). Key elements of the process as it relates to a computer program configuration item (CPCI) in a system program are illustrated in Figure 1-2. In addition to the specifications, documents shown in the figure include: (a) other technical documents such as handbooks and manuals which depend for their content on the computer program configuration; and (b) a set of special forms and reports involved in processing changes and reporting status. The actual CPCI is also represented, in the form of a magnetic tape symbol, as the eventual object of control; however, working procedures of configuration management are most directly concerned with the documents and forms.

It should be noted that the technical documents represented in the figure other than specifications--i.e., handbooks, manuals, plans--are important to configuration management because they are frequently subject to impact by changes to the specifications. Direct control of those documents is maintained by engineering and test functions; and they may undergo change for reasons unrelated to configuration management. However, configuration management is responsible for tracking and reporting their status, primarily in order to track the total implementation of approved changes to the specifications (see 5.2).

It may be inferred from the diagram that configuration management activities and procedures are implemented incrementally, beginning with baselining of the system specification and later expanding as specifications for individual items are completed and baselined. (Typically there are many of those, including equipment, although only one is shown.) Terms introduced for the successive baselines are:

- **Functional Baseline.** The functional baseline is defined by the system specification. In some programs, a set of lower-level specifications may be prepared in order to expand the performance and design requirements for major system segments. When this occurs, the functional baseline is defined by the entire set of system plus system segment specifications.
- **Allocated Baseline.** The set of approved performance-level (development, or Part I) specifications for configuration items is referred to as the allocated baseline. The expression derives from the principle that the development specification for each item is basically an expansion of the system performance and design requirements allocated to that item.

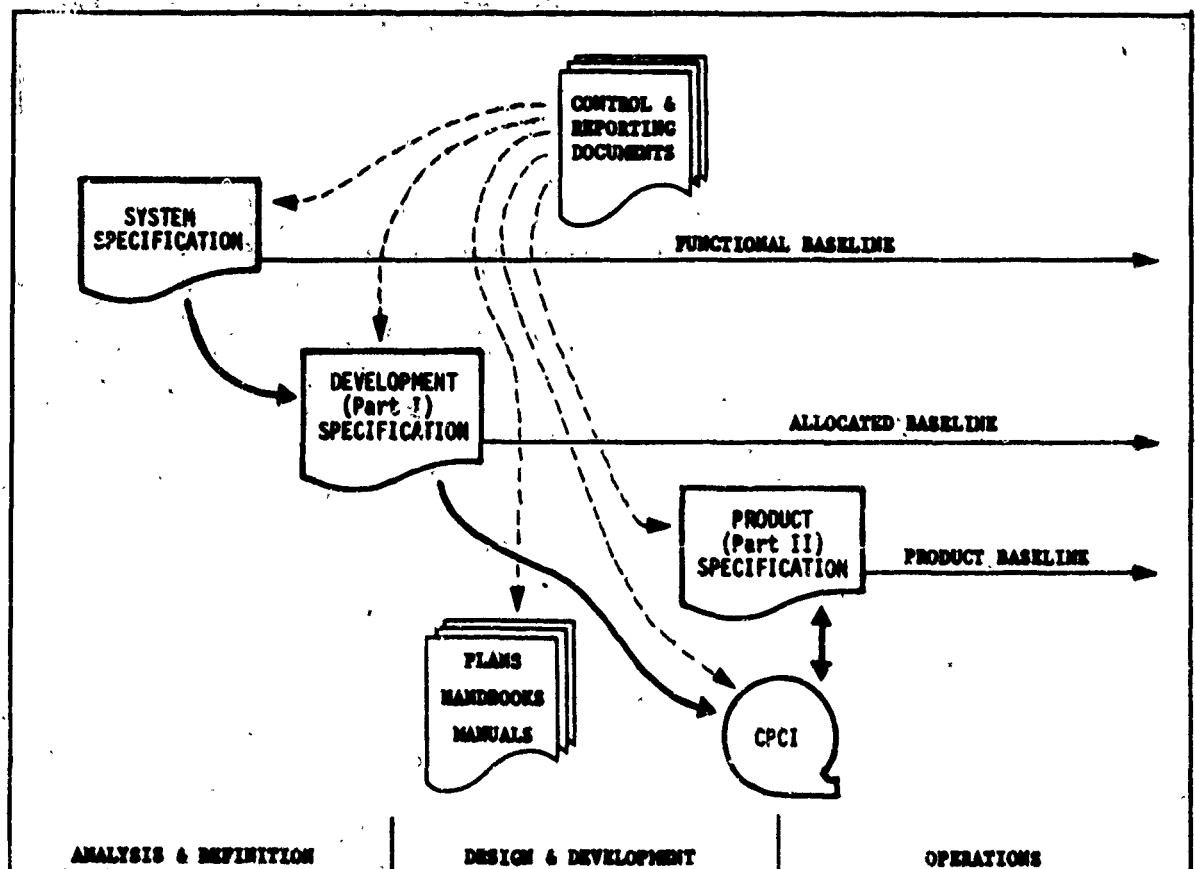


Figure 1-2. Sequence and Structure of Documents Involved in Configuration Management.

- **Product Baseline.** The product baseline is defined by the set of product (Part II) specifications for system configuration items, following their formal auditing and approvals.

The general process depicted in Figure 1-2 is not materially affected by the fact that development schedules for individual items in the system may be discrepant in their phasing. In system programs, it is also an important principle that all three baselines are maintained for the life of the system. Unlike practices which were once fairly common, it is fundamental to the present practice that an earlier, higher-level baseline is not discarded as a new one is added.

1.5 SOURCE REGULATIONS, SPECIFICATIONS, AND STANDARDS

The principal official documents dealing with various aspects of the information covered in later sections of this guidebook are listed in Table 1-1. Collectively, they provide comprehensive policy and requirements for software configuration management which have proved to be both sound and highly effective when properly applied and understood. Misuses and misunderstandings have been frequent, however, which can be attributed in part to such factors as the following:

- Except for some of the data item descriptions (DIDs), the documents are organized to address the configuration management discipline, rather than software as a separate system element. Predominant concern is with practices, procedures, and points of emphasis which are important primarily for hardware. Requirements specific to software are identified occasionally, but not consistently.
- Requirements under the various topics of identification, control, document maintenance, status reporting, and audits tend to be scattered among the many documents listed. Some are to be found only in the DIDs; most are expressed in directive language, with minimum explanatory guidance or cross-references to related requirements in other areas.

Coupled with those handicaps, the subject matter as a whole is inherently complex in its scope, potential depth, and interrelationships with other acquisition management activities. In attempting to alleviate those difficulties, this guidebook places emphasis on identifying the specific locations and nature of requirements in the various areas, and on explaining their intent, uses, and interrelationships. But it does not attempt to duplicate the source material itself. The content of later sections will be useful to software managers--and meaningful to other readers--to the degree that it is read and used in conjunction with direct knowledge of the referenced source material.

Table 1-1. Significant Source Documents Referenced in the Text.

Regulations/Manual

AFR 65-3	Configuration Management (Joint Services)
AFR 57-4	Retrofit Configuration Changes
AFR 800-14	Volume II, Chapter 6: Configuration Management
AFSCM/AFLCM 375-7	Configuration Management for Systems, Equipment, Munitions, and Computer Programs

Military Specifications/Standards

MIL-S-83490	Specifications, Types and Forms
MIL-STD-480	Configuration Control - Engineering Changes, Deviations and Waivers
MIL-STD-490	Specification Practices
MIL-STD-483	(USAF) Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs
MIL-STD-1521A	(USAF) Technical Reviews and Audits for Systems, Equipment, and Computer Programs

Data Item Descriptions

DI-A-3029	Agenda - Design Reviews, Audits and Demonstrations
DI-E-3101	System Specification
DI-E-3104	Addendum Specification
DI-E-3105	Inventory Specification
DI-E-3107	Installation Completion Notification
DI-E-3108	Configuration Management Plan (CMP)
DI-E-3118	Minutes of Formal Reviews, Inspections and Audits
DI-E-3119A	Computer Program Development Specification
DI-E-3120A	Computer Program Product Specification
DI-E-3121	Version Description Document (Computer Program)
DI-E-3122	Configuration Index (Computer Program)
DI-E-3123	Change Status Report (Computer Program)
DI-E-3128	Engineering Change Proposals (ECPs)
DI-E-3134	Specification Change Notice (Computer Program)

The fact that the source standards have been undergoing a number of changes in recent years is an additional complicating factor. Also, it happens that all of the documents cited most frequently herein for their software-related content are presently in the process of being revised, and in some cases of being reidentified. Those are:

AFSCM/AFLCM 375-7. This is the Air Force general configuration management policy and guidance document addressed primarily to in-house personnel. It is being revised and will be reissued as an Air Force document other than a manual--probably as a pamphlet--and will bear a different number.

MIL-STD-483 (USAF). This is the Air Force supplement to the DoD configuration management standards, which now contains most of the contractually-applicable requirements for configuration management of computer programs. Some parts of it are being revised for incorporation into a DoD-level standard, presently identified in draft form as MIL-STD-XXX. Whether MIL-STD-483 will continue to exist as an Air Force supplement is not yet known.

MIL-STD-480. Revisions may eventually incorporate some software change processing requirements presently contained in MIL-STD-483. Present plans are to issue an interim MIL-STD-480A, pending coordination of additional revisions.

MIL-STD-490. The revision will incorporate format/content instructions for computer program specifications presently contained in Appendix VI of MIL-STD-483, together with other changes. The revision will be identified as MIL-STD-490A.

Firm schedules for actual issuance of the approved revisions are not yet available. When they do appear, a revision of this guidebook will be indicated in order to take account of the changes. Based on review of coordination drafts issued to date for all of those documents, however, it appears that the impact on software requirements will be primarily on their locations, rather than on their content.

1.6 SPECIAL TERMS

Formal definitions of terms and abbreviations are provided in the glossary, Section 9, for purposes of reference. The comments below address a selected few terms which are particularly important to the subject matter of later sections, but which have been used with varied and often misleading connotations. Purposes of these comments are to explain the intended meanings of the terms as they are used herein, and to identify the nature of certain ambiguities.

1.6.1 Computer Program

A computer program is generally understood to be a sequence of coded instructions, including coded values for fixed elements of data, designed to cause an assembly of computing equipment to perform a function or set of functions. While there are some uses of the term which imply an instruction sequence of limited size or complexity, no such limitation is implied herein. The term refers to any set of instructions (presumably coherent), of whatever size or complexity. A computer program may be a CPCI, or part of a CPCI, or it may not be designated as a CPCI (see 1.6.2 below).

While a computer program may include certain elements of coded data, it may not consist wholly of data. For example, a magnetic tape containing only coded input data values for insertion into an automatic test and checkout equipment is not a computer program. This distinction is obviously subject to certain problems, which have resulted in controversies and conflicting treatments in current Air Force/DoD documents. Pending clarification, questions arising must be examined and resolved on a case-by-case basis.

1.6.2 Computer Program Configuration Item (CPCI)

A CPCI is any computer program which satisfies an end-use function and is specifically designated by a controlling agency for configuration management. Not all computer programs used during the course of a project are necessarily designated as CPCIs, e.g., ones which may be generated and used solely for development and test purposes. However, it is Air Force/DoD policy that those being developed in a given program for delivery to the procuring activity are to be designated and managed as CPCIs.

CPCIs are identified in each program on the basis of criteria which are discussed further in the next section. A CPCI may be very large or very small, depending more on management than on technical considerations. That is, the determination that a given assembly of code constitutes a CPCI is based heavily on such factors as source, whether developed or bought, schedules, and eventual use and control.

A CPCI is the actual computer program end item in the form of coded instructions recorded on a medium (tape, cards, disc) suitable for insertion into the computer. As such, the CPCI does not include the specification, since the specification is a separately-deliverable item of contractor data.

1.6.3 Computer Program Component (CPC)

A CPC is a major part of a CPCI, identified for purposes of convenience in specifying and developing a CPCI as a set of subordinate elements. CPCs are normally understood to constitute the first-level breakdown of an item as specified in its Part II specification; i.e., they are the set of next-smaller computer programs that make up the CPCI as a whole. However, CPCs have been identified at a somewhat lower assembly level, for a very large and complex CPCI, when necessary for their adequate technical description in the Part II specification.

Thus, CPCs are structural parts of the end item. They may or may not correspond individually with major functions of the CPCI which are specified in its Part I specification.

1.6.4 Engineering

"Engineering" is used in this guidebook in the broad sense of its definition and use in such documents as AFR 800-3 and MIL-STD-499A, referring to any or all of the various lines of technical effort involved in a system program. It is the general term which encompasses system engineering, as well as the many equipment engineering specialties, software engineering, and human factors engineering. Within that broad concept, further distinctions observed in the text are as follows:

- Component Engineering is the general term for any specialized branch of engineering, in which the primary focus of analysis and design activities is within the scope of a given technical field or on one class of system components, e.g., electrical, electronics, communications, or software.
- System Engineering is characterized by its focus on levels of analysis, design, interface control, or other integrating activities which cut across some number of component engineering disciplines.
- Software Engineering refers to the specialized technical knowledge and effort required to design, develop, implement, test, evaluate, and support computer programs.

1.6.5 Software

As used herein, "software" is completely synonymous with "computer program(s)". Because of widely conflicting variations in its established meanings, the use of this term has been carefully avoided in current Air Force configuration management standards. It is recommended that contractual uses of the term be confined to cases in which it is clearly defined, in the contract, to be equivalent to "computer program(s)". As a separate class of deliverable end items, software (computer programs) should not be construed as including contractor services, the specifications, or other items of associated documentation deliverable against the CDRL. (See also the note, 7.2 herein, which reviews some recent questions raised by uses of this term in DoDD 5000.29.)

SECTION 2. SELECTION OF COMPUTER PROGRAM CONFIGURATION ITEMS

The selection of configuration items is a process which normally occurs during early stages of a system program. Simply stated, it is the process by which the complete set of equipment, computer programs, and facilities elements contemplated for a system as a whole are separated for purposes of managing their development or other procurement into individually-identified subsets. Air Force policy underlying the configuration item concept has been summarized succinctly in the following terms:

"...systems/equipment are not procured by single identifiable systems but rather by separate end items of contractor peculiar items, Air Force Supply Federal Stock, and commercial 'off-the-shelf' items."*

Hence, the configuration item is regarded as a level of management. Specifically, it is the level:

- At which the procuring activity specifies, contracts for, and accepts individual parts of the system.
- Below which the developer is responsible for management of the development, or procurement, and assembly of item components.
- Above which the procuring activity retains responsibility for interfaces, integration, and system performance.

2.1 REQUIREMENTS AND CRITERIA

Basic principles governing the selection of CIs in general, including a few criteria specific to computer programs, are set forth in paragraphs 1-17 through 1-21 of AFSCM/AFLCM 375-7. However, it is an important point of emphasis, throughout that source, that the selection process is not subject to "stylized" rules. Decisions should be based on experience, knowledge of the principles and implications, knowledge of the given system program, and attention to both technical and administrative considerations.

The identification of a given assembly of computer program instructions and coded data as a CI is basically a technical product of the system engineering process. Although accomplished at an early stage of the program, it represents

*AFSCM 375-1, "Configuration Management During the Acquisition Phase", 1 June 1962; p. II-3.

a design decision, resulting from the steps of: (a) functional analysis and definition of system performance requirements, and (b) system design, during which the defined functional and performance requirements are allocated among planned assemblies of system physical elements. Sufficient analysis and study of computer program design at the system or system segment level must be performed to assure the technical soundness and feasibility of the to-be-developed CPCIs. At that early stage, the CI designation constitutes a commitment to develop a deliverable end item--e.g., in the form of a tape or deck of cards--which will perform its allocated functions when eventually assembled into the system.

The assembly levels to be identified as CIs are not arrived at, however, solely through technical considerations. In a system program a significant responsibility of engineering managers is to plan and direct the technical analysis and design effort in such a way that the proposed levels of assembly selected as CIs meet established criteria for their subsequent management. At the outset, for example, system engineering studies resulting in CI selection must be guided by Air Force policy that computer programs are to be managed as configuration items (AFR 800-14), and that computer programs are not to be identified as components of equipment CIs (AFSCM/AFLCM 375-7). Other major requirements and criteria to be observed in the process of selecting CPCIs are summarized below.

2.1.1 Requirements

The CI (originally "contract end item") is a level of assembly which the program office procures from a single contractor or other source. It is the level at which a program office exercises formal management control over the responsible contractor in the areas of configuration management, procurement, program control, and monitoring of the contractor's technical progress. In planning and implementing the system program, the following documents and actions apply separately to each CPI:

- Specifications.
- Proposed engineering changes, and reports of change implementation.
- Management information reporting against the contract work breakdown structure.
- The performance of technical reviews and configuration audits--PDR, CDR, FCA, and PCA.

- The preparation of operating and user manuals.
- The developer's formal test program.
- Formal acceptance by the procuring activity.

2.1.2 Selection Criteria

Criteria listed below are to be regarded as a "shopping list", in that both the importance and applicability of the considerations listed vary widely among different system programs. The fact that the criteria are not independent of each other points up, further, the need for careful consideration of all relevant factors which apply to each CPCI.

In all cases, the intended source (contractor) is an essential starting point for decisions, since (a) assemblies of computer program elements to be acquired from a single contractor are potentially a single CPCI, and (b) assemblies to be acquired from separate sources must be separate CPCIs. Factors of cost, complexity of documentation, interface control, and other requirements identified above dictate that it is generally desirable to avoid having any more CPCIs than necessary. Hence, for a given single contractor, a productive approach is to start with the tentative assumption of a single CPCI, then "shredout" into separate CPCIs only when fully justified by an applicable criterion.

- Separate Computers. Computer programs to be designed for operation in different types or models of computers must be separate CPCIs.* Separate CPCIs may also be indicated when a given installation uses a number of computers of the same type/model, each performing different functions in the system as a whole and having different sets of interfaces with other system elements.
- Separate Schedules. Computer programs scheduled for development, testing, or delivery at different times may be separate CPCIs. When indicated by interrelationships and intended use, however, consideration should be given to such alternatives as: expansion of the earlier-developed CPCI via ECP; or development of the later CPCI to incorporate and replace the earlier item.

*By definition, the CPCI must be "in a form suitable for insertion into a computer". If a single computer program, in the form of assembly code, happens to be fully compatible with the characteristics of more than one type or model of computer--and can be so qualified--that condition would be satisfied.

- Different System Functions and Uses. In general, mission, support, and diagnostic (off line) computer programs should be separate CPCIs. Consider: intended locations of use, expected change cycles, and user personnel directly concerned with their functional and performance characteristics, together with related responsibilities for deployment phase control (see below).
- Different Deployment Phase Control. Computer programs intended for different systems,* and/or for different configuration control during the deployment phase, should be identified as separate CPCIs, even though they may be largely identical at the time of initial development and delivery. Consider planning for user command(s) and AFLC deployment phase control documented (or to be documented) in the Computer Resources Integrated Support Plan (CRISP) and Operational/Support Configuration Management Procedures (O/S CMP).**

2.2 PITFALLS

Although a single "right" solution may not always present itself, reasonable care and attention to the considerations outlined above should yield sound results. On the other hand, because of the importance of the CPI selection step to all subsequent phases of a program, success of the program can be almost precluded if those objectives and principles are disregarded. Examples of relatively prominent misconceptions which have led to serious difficulties in recent programs are summarized below.

- Development Specification vs. the CPI. System and system segment specifications have been placed on contract which were incomplete with respect to CI/CPI selection and functional allocations, with the requirement for delivery of Part I CPI specifications within a short time after contract award, and with no requirement for the contractor to perform (or document

*The reference is to CPCIs designed to perform mission functions. Standardization of CPCIs for broad application is a more important and realistic objective for those support computer programs which depend more for their nature and usefulness on the computer equipment than on the operational mission.

**See Volume II of AFR 800-14 for discussions of the CRISP (Chapter 3) and O/S CMP (Chapter 6). The CRISP must include the assignment of control responsibilities for computer programs during the deployment phase. The O/S CMP further details the planning and procedures. In effect, control may transfer to the supporting command (AFLC) for some CPCIs, and to the using command for other CPCIs of the same system.

and deliver) system engineering analysis and design studies as a basis for CPCI selection. Under those circumstances, "CPCIs" have been selected and identified on the basis of system functions alone, without benefit of adequate system or segment-level computer program design studies to verify their feasibility or cost-effective development as separate computer programs. In one case, approximately 10 such "CPCIs" as identified in their Part I specifications eventually had to be combined into one massive computer program and documented in a single set of specification data at the product level. While such a case clearly involves a complex of errors, it appears that one important element of the problem lies in the failure to appreciate relevant distinctions between the CPCI and its specification, particularly at the Part I level. Specifically, the CPCI selection question is being approached in some instances from the point of view of how to sort out system functions into Part I specifications, rather than from the point of view of how to allocate them to deliverable computer program end items (see 2.1 above).

- Size and Visibility. Coupled with the misconception noted above is the assumption that breaking down a complex of data processing functions into a number of separate CPCIs makes the elements more manageable, and more "visible" for purposes of technical monitoring. However, neither size nor visibility is consistent with the accepted criteria for selecting either computer program or equipment CIs (except that size has been applied in the reverse manner, to avoid having large numbers of small CIs). While one small item is generally easier to manage than one large one, the total management task is necessarily increased if the large one is broken down. If technical management procedures are carried out at the proper level in both cases, the increase in number of CPCIs is more likely to hamper visibility than to improve it. Undesirable results include:

- Paperwork involved in preparing, processing, and status reporting of engineering changes tends to be multiplied by the number of CPCIs.
- The burden of maintaining interface control can be amplified significantly if operating interrelationships exist among the separated items.
- The CPCI development time and costs, together with resulting total size and operating times, can be increased.

In effect, the argument for using size and visibility as selection criteria is really the same, in some respects, as the argument that any large contract should be broken down into a number of smaller ones. In both cases, the only true result is a net increase in management effort, overhead paper, and difficulties in maintaining coordination.

2.3 THE TYPE CLASSIFICATION

In some systems, a given CPCI is designed for use at a number of system site locations but must be adapted to the operating environment at each location. Typically, the adaptation is accomplished at the time of installation through incorporating coded data values (adaptation data) appropriate to each location, as a part of the CPCI's fixed data base. Alterations of the computer instructions in the form of adding, deleting, or modifying processing capabilities may also be involved. In such cases, there are two alternatives to consider:

- Identify the computer program to be used at each location as a separate CPCI. As indicated by the circumstances, the further option should be considered of whether to (a) prepare complete separate specifications or (b) use the addendum specification (see 3.5).
- Classify the different configurations, including adaptation data, as types within a single CPCI. In this case, only one specification is prepared; but the types are listed in the "Scope" statement and each type is further specified throughout the specification in accordance with instructions of MIL-STD-490, paragraphs 4.1.2 and 4.3.b.*

The latter alternative is recommended in all cases when variations are confined to adaptation data, but should also be considered if the differences in computer instructions are minor, since the potential savings in management costs can be substantial.

At the same time, the development and management must be accomplished in such a way that the exact configuration at all locations is fully specified, controlled, and known. If types within a CPCI are proposed, the contractor's configuration management plan should include explicit treatment of how the types will be handled in such areas as the specification, change proposals, specification maintenance, status reporting, and configuration audits.

2.4 IDENTIFICATION NUMBERS AND MARKINGS

Numbers are employed to identify configuration items, parts, specifications, other documents, and special forms associated with configuration management. In general, some numbers may be assigned directly by the procuring activity, although most are assigned by contractors in accordance with prescribed standards. A "number" consists typically of a specified maximum or exact number of alphabetic and/or numeric characters.

*This treatment applies to both Part I and Part II of the specification.

The fact that numbers must be assigned to identified documents and items is specified in Appendix IX of MIL-STD-483. However, the specific requirements are to be found in various other places. Table 2-1 identifies certain classes of numbers and markings which are of interest to software configuration managers (and to data managers) and identifies the direct sources for their requirements, where those exist.

Table 2-1. Sources of Requirements for Numbers and Markings

ITEM	SOURCE	REMARKS
SPECIFICATIONS	MIL-STD-490 para. 3.2.16	See also 5.1.2 herein
OTHER DOCUMENTS	-	*
CPCis	MIL-STD-483 " 90.3.2.3	Exactly 7 characters
CPCs	AFSCM/AFLCM 375-7 " 1-39,1	
ECPs	MIL-STD-480 " 10.6	
CLASS II CRs	-	*
SCNs	MIL-STD-490 " 3.3.2.3	See also 5.1.3 herein
VERSIONS	MIL-STD-483 " 80.12.1.1f	See also 5.4.2 herein
VDDs	AFSCM/AFLCM 375-7 " 1-39h(3)	See also 5.4.2 herein
CONFIGURATION INDEX	MIL-STD-483 " 80.10.2	Issue number only
ACTIVITY CODE	Handbook H 4-1	
SECURITY MARKINGS	DoDD 5220.22-M	
CARDS, TAPES, etc.	-	*
*Required, but uniform standards not specified.		

Closely related to numbers are standards for marking and/or coding of identification data on magnetic tapes, canisters, card deck containers and cards, or other storage media for computer program code. As the table indicates, uniform requirements in those areas are not prescribed in the current configuration management standards. Hence, any particular requirements which the procuring activity may have in those areas must be spelled out directly in the contract. Appropriate coverage of this topic within his set of internal standards (see 4.1) should also normally be identified in the contractor's configuration management plan.

Numbers and other identification data pertaining to maintainable documents, which tend to be matters of key importance to software configuration managers, are discussed further in Section 5 of this guidebook.

Requirements for a centrally-controlled "computer program identification number (CPIN)" are mentioned in Volume II of AFR 800-14, and are presently in process of being further developed within AFLC. AFLC Supplement 1 to AFR 800-14, Volume II, indicates that the CPIN (approximately 15 characters) will be used to identify not only the computer programs but also their specifications and associated documents of a developmental or test nature, whereas user documents (handbooks, manuals) will be managed as technical orders (TOs). If adopted as outlined, that system promises to have a number of potentially-confusing impacts on currently accepted practices of AFSC POs, and perhaps also of the using commands. Information regarding actions that may be taken to resolve the various questions which it raises is not yet available, however.

SECTION 3. SPECIFICATIONS

This section identifies the source of Air Force requirements governing specifications, and summarizes the nature, functions, and applicability of specifications that may apply to computer program contracts during a system acquisition. The three references of primary interest are:

- MIL-S-83490, "Specifications, Types and Forms", 30 October 1968. MIL-S-83490 is a relatively small (5 page) military specification which defines a uniform structure of types, subtypes, and forms of specifications that may be developed by either Government agencies or contractors for the acquisition of military systems, equipment, computer programs, or materials.
- MIL-STD-490, "Specification Practices", 30 October 1968. MIL-STD-490 contains the detailed standards for format, content, and maintenance of specification types and subtypes identified in MIL-S-83490. It is organized into a basic standard containing provisions that apply to specifications in general, with a series of 15 appendices devoted to format/content requirements for individual specification types and subtypes. Table 3-1 lists those by type number, title, and the MIL-STD-490 appendix number.

Table 3-1. MIL-STD-490 Appendices for Specification Types and Subtypes.
(Asterisks identify subtypes that may apply to computer programs.)

	<u>Appendix</u>
Type A - SYSTEM SPECIFICATION	I
Type B - DEVELOPMENT SPECIFICATION	
Type B1 Prime Item Development Specification	II
Type B2 Critical Item Development Specification	III
Type B3 Non-Complex Item Development Specification	IV
Type B4 Facility or Ship Development Specification	V
*Type B5 Computer Program Development Specification	VI
Type C - PRODUCT SPECIFICATION	
Type C1a Prime Item Product Function Specification	VII
Type C1b Prime Item Product Fabrication Specification	VIII
Type C2a Critical Item Product Function Specification	IX
Type C2b Critical Item Product Fabrication Specification	X
Type C3 Non-Complex Item Product Fabrication Specification	XI
*Type C4 Inventory Item Specification	XII
*Type C5 Computer Program Product Specification	XIII
TYPE D - PROCESS SPECIFICATION	XIV
TYPE E - MATERIAL SPECIFICATION	XV

- MIL-STD-483 (USAF). "Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs", 12 April 1971. This is the Air Force configuration management standard which supplements the DoD standards in special areas pertaining to systems, equipment, and computer programs. Complete content instructions for the computer program development and product specifications (Types B5 and C5) contained in Appendix VI of MIL-STD-490 normally replace, for Air Force acquisitions, those of MIL-STD-490 referenced above. The only current source of requirements for the addendum specification form (see 3.5) is Appendix IV of this standard.

3.1 THE SPECIFICATION TREE/CI LIST

The term "specification tree" derives from an earlier engineering practice of identifying specifications at levels corresponding to levels of item assembly, or installation, into a system. The engineering levels may still vary widely, for equipment items, in that assemblies designated as CIs may range from parts as small as an altimeter to an entire aircraft. However, under current concepts of uniform specifications, all CIs are regarded as being at the same level--i.e., the CI level--for purposes of configuration management. When so depicted in the specification tree, the result is essentially equivalent to a CI list. Both the specification tree and CI lists are approved forms for identifying computer program and equipment CIs in the system specification (ref. paragraphs 30.2 and 30.3, MIL-STD-483). For a system as a whole, the maximum number of specification levels that may be identified is three. Those are depicted in Figure 3-1 and explained briefly in the following subparagraphs.

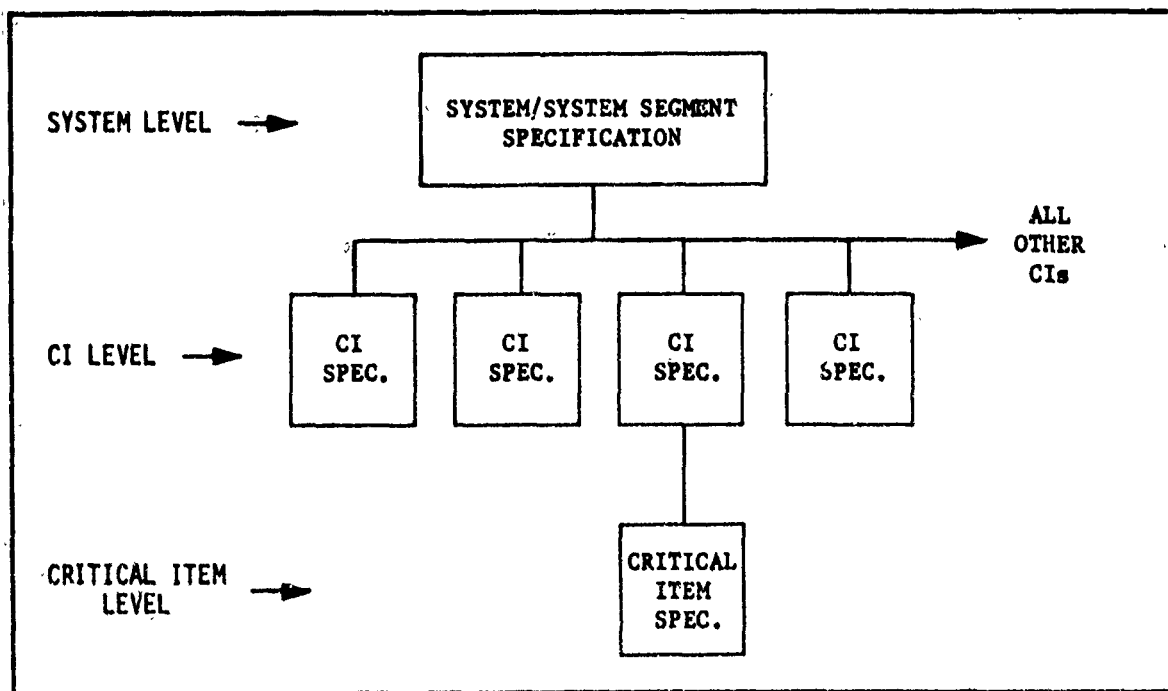


Figure 3-1. The Specification Tree

3.1.1 System/System Segment Specification Level

Each system is specified by one (Type A) system specification prepared in accordance with Appendix I of MIL-STD-490. When system segments are identified, the system specification may be structured into volumes, one covering general requirements for the system as a whole and a separate volume for each system segment (ref. Appendix III, MIL-STD-483). Thus, the set of volumes constitutes a single specification.

As the terms are used in the Air Force, "system segment" is closely related to "subsystem". However, the system segment concept refers more directly to a part of the developmental program, rather than to a functional/physical part of the resulting system. Developmental responsibilities for a system segment are analogous to those of a configuration item, in that responsibility for an identified system segment may be assigned to only one contractor or Government agency. Each system segment normally includes some number of equipment and/or computer program CIs, together with associated requirements for system and human factors engineering and for such tasks as developing, documenting, testing, and assembling the configuration items. Major sets of operational and support computer programs have been the basis for identifying separate system segments in some C³ system programs.

The term "functional area" is used in MIL-STD-490 to refer to the first-level breakdown in the system specification. However, the designation of what constitutes a functional area is flexible. In Air Force use, a functional area is normally equivalent to a system segment in the general volume of a system specification, but refers to the next-lower level of assembly in each volume devoted to a system segment.*

3.1.2 CI Specification Level

Each configuration item is specified by a single specification. In terms of the types listed in Table 3-1 above, that specification may be composed of only one type or a combination of two types, as follows:

- Single Type. A given item may be specified entirely by only one specification type (or form; see 3.5.1) if the applicable specification is: Type C4 (for inventory items); Type C1a or C2a (for equipment CIs selected on a "form, fit, and function" basis); or Form 3 (commercial practice). Applicability of the Type C4 and Form 3 specifications to computer programs is discussed further in 3.5 below.

*A further discussion of questions pertaining to functional areas vs. system segments is provided in 7.3.

- Combined Types. For items being newly developed in a given system program, the development (Type B) and product (Type C) specification types are normally combined to form a single, two-part specification having a single specification number and designated Part I and Part II., respectively. Combinations to which this practice applies are:

B1/C1b	-	Prime Item
B2/C2b	-	Critical Item
B3/C3	-	Non-Complex Item
B5/C5	-	Computer Program Item

Thus, specifications are referred to by different labels which tend to be interchangeable. For computer programs:

- The terms "computer program development specification", "Type B5 specification", and "Part I CPCI specification" are interchangeable; and
- The terms, "computer program product specification", "Type C5 specification", and "Part II CPCI specification" are similarly interchangeable.

3.1.3 Critical Item Specification Level

A critical item (formerly "critical component") is a special subassembly within an equipment CI which may be designated as critical for engineering or logistics reasons and controlled by its own separate specification. The critical item designation does not apply to computer programs. Hence, the specification tree for a system and its entire collection of computer program CIs is really limited to only two levels.

3.1.4 Specification Tree vs. Generation Breakdown

The specification tree (CI list) is a direct result of the CI selection process discussed in Section 2 above. Although CI/specification numbers and specification types are supplied by configuration managers, the identifications of equipment and computer program assemblies to be designated as CIs represent essential steps in system design. From the designers' point of view, the selection criteria are often perceived as arbitrary constraints which do not clearly contribute to the efficient accomplishment of that process. And in fact, they may not; it is mainly to be hoped that they will not unduly complicate it by creating interface problems or forcing premature design decisions.

Whether hardware or software, it is established engineering practice to perform the analysis and design process over a developmental period in a series of "top down" steps. Once the system is designed at the level of identified sub-systems and CIs, further breakdowns into assemblies, subassemblies, etc. are then identified and documented at successively lower levels down to the level of transistors, bolts, or single computer instructions. The "tree" resulting from that process is known in equipment engineering as a generation breakdown structure. It may also be known as an assembly or installation tree, referring to its later function as a roadmap to the manner in which parts are installed in the system.

The existence of that concept in software is evidenced by the various labels that have been attached to the different assembly levels, e.g., system, sub-system, program, subprogram, module, routine, et al. However, the generation breakdown structure does not have the many uses for computer programs that it has for equipment items, due to the absence of requirements for eventual manufacture and supply of subassemblies and parts. Configuration management requirements in this area for software are largely confined to only two levels below the CPCI level itself, which are identified in very general ways as (a) computer program components (CPCs) and (b) the computer instructions.* Those are recognized by configuration management as levels which should be included in the structure of almost any CPCI and for any chosen approach to its technical design. Identification and control at additional levels during the development process are matters primarily of technical, rather than configuration management, responsibility and concern (see also 4.5.1).**

*That is, referring to structural characteristics of the computer program as described in its Part II specification. In the Part I, a related but different breakdown exists in terms of functions and subfunctions.

**"Top-down programming" refers to a given design/development approach wherein CPCs or modules are arranged in a hierarchy on the basis of their control and sequencing; like other design approaches, it affects the content of information dealt with in the specifications, but has no effect on configuration management.

3.2 SYSTEM SPECIFICATION

Primary sources of requirements for the format and preparation of the system specification are:

MIL-STD-490:

Appendix I. Type A, System Specification

MIL-STD-483 (USAF):

Appendix III. System Specification/System Segment Specification

The latter source supplements Appendix I of MIL-STD-490 with respect to preparation of the specification tree and CI list, use of the Type A specification form for system segments, and the inclusion of selected information pertaining to computer programs.

3.2.1 Content

As is true of most other specifications, "the" system specification may consist of a collection of separate documents, both because: (a) it may be prepared as a set of separate volumes (see 3.1.1 above); and (b) information may be incorporated by reference to system engineering documentation or to specific requirements set forth in other applicable documents.

The primary purpose of the system specification is to define requirements at the level of system functions and performance. However, it also serves the important function of specifying requirements for system-level design, in that it identifies and allocates system functional/performance requirements to system segments and configuration items (i.e., when completed; see 3.2.2 below). The completed specification contains the following principal elements:

- Identification of the general system configuration, in terms of system segments and/or functional areas.
- Definitions of performance and design requirements and constraints for the system as a whole, and allocations of those to the functional areas.

- Identification of operational and support configuration items to be developed or otherwise acquired for each segment/functional area, including computer equipment, consoles, peripherals, communications, and computer programs.
- Definitions of functional interfaces among system segments/functional areas and of the system as a whole with external systems.
- Descriptions of relevant organizational, operational, facilities, maintenance, and personnel and training concepts.
- Specification of system test requirements, in terms of methods of verifying the specified requirements for system performance and design.

3.2.2 Development and Control

Generalized steps involved in developing and completing a system specification are depicted in Figure 3-2. During the conceptual phase: basic requirements are derived from the major command's statement of a required operational capability (ROC); alternative system concepts are examined for potential mission effectiveness and feasibility; a firm system concept is selected, leading to an initial program management directive and activation of the PO cadre; and system engineering studies are conducted to expand the operational requirements into criteria for system performance and design.

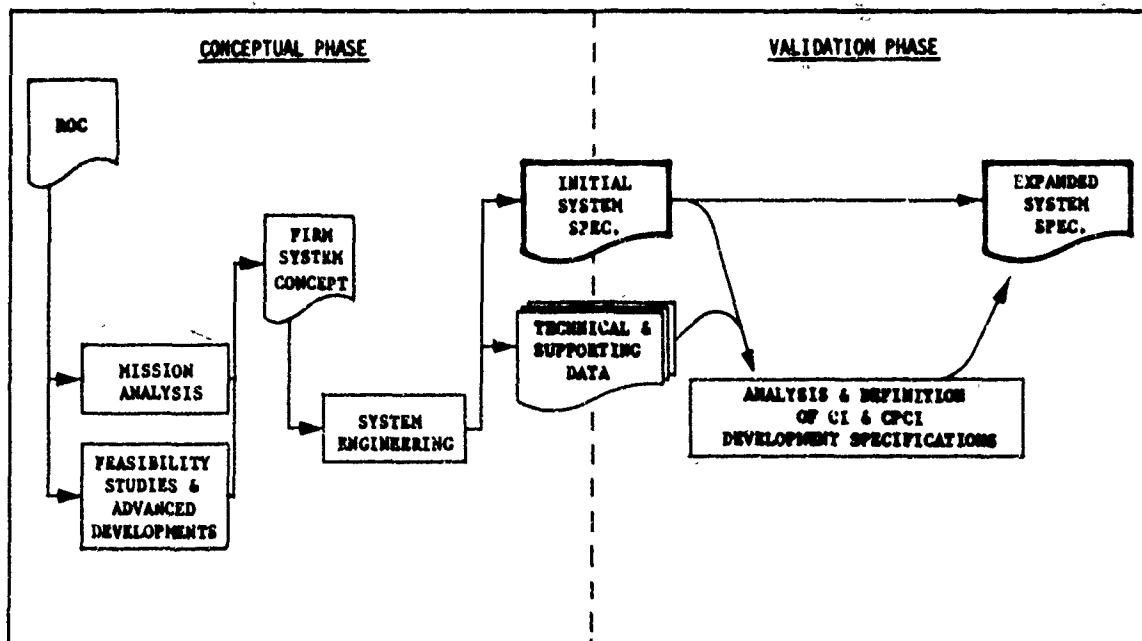


Figure 3-2. Development of the System Specification

Those tasks are accomplished by in-house Air Force capabilities, using not-for-profit and other contractor support where appropriate. The primary technical product of the conceptual phase, the initial system specification, is authenticated and established as the system functional baseline through in-house procedures of review and approval, at the outset of a validation phase. It is normally not complete at that stage, leaving specific portions to be accomplished by the validation phase contractor(s).*

During the validation phase, further detailed system and component engineering studies are performed to: expand operational and support functions; perform trade-off, feasibility, and risk studies to identify equipment and computer program CIs; allocate system/system segment and functional area requirements to CIs; and prepare CI development (Type B) specifications based on further detailed expansion of the allocated requirements. While other revisions may also be indicated, specific portions of the system specification to be completed as a result of those studies are:

- A complete list of system computer program and equipment CIs, including commercial and Government inventory as well as developmental items, is to be provided in paragraph 3.1 (System Definition). This list is organized in numerical sequence by CI numbers.
- The specification tree for the system is to be delineated in paragraph 3.1.4 (System Diagrams), including specification numbers for all items shown in the tree.
- Definitions of functional interfaces among system segments/functional areas are specified in paragraph 3.1.5 (Interface Definitions), either directly or by reference to ICDs (see 4.6).

Those and all other changes to the system specification which are accomplished during the validation phase, or at any later time in the system program, are processed through formal procedures of configuration control and specification maintenance. The CI list and specification tree should be firmly defined at the time of the SDR, including identified commercial and Government inventory items as well as CIs and CPCIs to be newly developed for the system.

3.3 COMPUTER PROGRAM DEVELOPMENT SPECIFICATION

The computer program development (Part I, or Type B5) specification is the document which states requirements for the development of a computer program CI in terms of functions, performance, design constraints, and tests/verifications required to demonstrate that those characteristics are achieved.

*Certain conflicting statements on this topic appear in AFR 800-14; see 7.4.

The applicable data item description (DID), DI-E-3119A, directs preparation of the specification in accordance with instructions contained in Appendix VI of MIL-STD-483. Instructions for structuring the specification for a complex CPCI into a series of separate volumes are not included in Appendix VI, but are provided in Section 10 of the DID itself.

The DID for the development specification is placed on validation phase and full-scale development phase contracts to govern (a) initial preparation of the specification and (b) the subsequent preparation of change pages or revisions resulting from approved engineering change proposals. The description below assumes that initial preparation occurs during the validation phase. For complex mission CPCIs in particular, a validation phase or equivalent effort is normally required to achieve a level of completeness and accuracy which is adequate as a basis for initiating full-scale development.

3.3.1 Content

The computer program Part I specification is primarily a detailed definition of performance-oriented data processing requirements to be met by the CPCI when developed. It is written in operational and logical language to define precisely all data and processing tasks of the CPCI, including accuracies, data volumes/frequencies, and other related requirements. Provisions are expressed in directive terms and addressed to the computer program developer (contractor), since its immediate intended role is to serve as an important part of the developer's contract. When completed, its technical content consists of the following principal elements:

- Identifications and detailed definitions of all interfaces with other equipment and computer program CIs.
- A description of the operational functions and subfunctions to be performed by the CPCI.
- Definitions of all specific input, output, and processing requirements for each function/subfunction, including data definitions for elements of the data base.

- Design requirements and constraints, in terms of computer programming languages or design standards.*
- Specification of methods and levels of DT&E by which required performance and design characteristics of the developed item are to be verified.

Thus, the significant emphasis in the development specification is on providing a comprehensive definition of the CPCI configuration at the level of its required functional characteristics as a part of the system. While it includes essential design requirements and constraints, it avoids specifying the design as such. For example, the "functions" for which input, output, and processing requirements are specified are derived through successive expansions of system functions; they do not dictate structural components of the eventual CPCI.**

3.3.2 Development

Figure 3-3 illustrates how the Part I specification development for a complex mission CPCI should relate to other events and efforts during a "model" validation phase. Aspects of the engineering process which are significant to configuration management and the subsequent software acquisition include the following:

- System engineering studies should result in the selection of equipment and computer program CIs at about the time of SRR. The SRR emphasizes review of system engineering analysis data to support the developer's convergence on an optimum and complete configuration (Appendix A, MIL-STD-1521A).
- Firm identifications of CIs and allocations of system functions are essential prerequisites to initiating the development of Part I specifications.

*MIL-STD-483 (para. 30.5) also provides for similar requirements to be stated in paragraph 3.3.8 of the system specification. In practice, that portion of the system specification should emphasize requirements which apply to all system CPCIs and can be specified by reference to existing standards, whereas the Part I CPCI specification (in para. 3.2.n) applies specifically to the given CPCI; the latter may specify design requirements by reference to paragraph 3.3.8 of the system specification and/or add others not covered therein.

**A first task in computer program preliminary design (later, to be completed prior to PDR) is to allocate the development specification functions to CPCs; ref. paragraph 30.2.2a of MIL-STD-1521A.

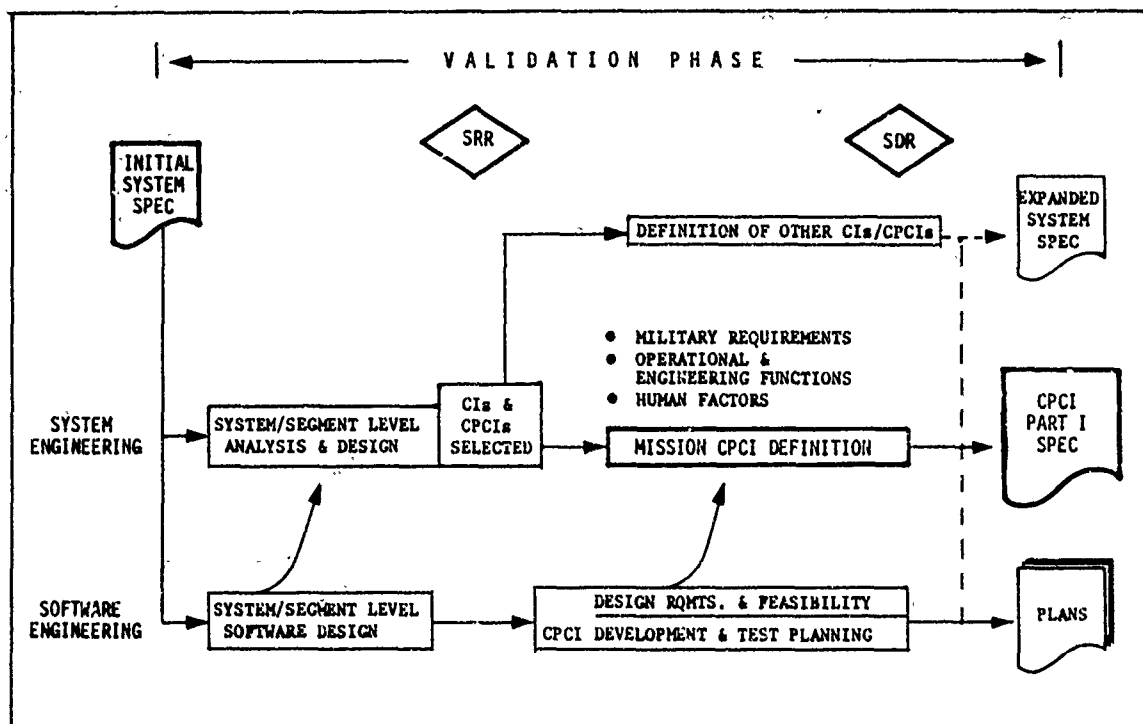


Figure 3-3. Development of the CPCI Part I Specification

- The Part I specification development for mission CPCIs is a system engineering task, as distinct from software engineering, since its essential orientation is towards system operational functions, including human factors.*
- The role of software engineering as such in supporting the Part I specification development consists of verifying design feasibility of the proposed functional and performance requirements, inputting design requirements, and participating in the definitions of Section 4 test requirements. CPCI-level design required to provide that support is not documented in the Part I specification, however. Design, timing, and sizing studies may be documented separately; but their results should also be directly visible in the computer program development plan (CPDP), which represents a major end product of software engineering during this phase.

*System engineering responsibility is the rule for CI development specifications in general (ref. Appendix B of MIL-STD-881A). However, the predominant required knowledge may be in a component equipment or software engineering field in the case of some items, e.g., for maintenance-diagnostics or compilers.

3.3.3 Approval and Functions

The verification of development specifications for completeness, accuracy, and compliance with requirements does not involve a formal configuration audit as it does for product (Part II) specifications. Chapter 2 of AFSCM/AFLCM 375-7 outlines the development specification review, approval, authentication, and baselining procedures for which the procuring activity's CMO is responsible at the end of a validation phase, including certain contingencies. Formal delivery of the approved specification is accomplished by the contractor following in-house specification review, resolution of comments, and receipt of the program manager's authenticating signature on the cover page. To the contractor, the specification is effectively baselined for formal configuration control when it is incorporated into his full-scale development contract as a compliance document.

Criteria for evaluating detailed format and content of a completed Part I CPCI specification are subjects to be expanded in the Requirements Specification guidebook. Summarized briefly, major functions of the document to be kept in mind in the course of both technical and configuration management evaluations are as follows:

- The Part I specification functions as the procuring activity's key contractual compliance instrument to govern computer program acquisition. It is the only CI-level specification which ever serves that purpose (see 3.6).
- When written in accordance with format/content instructions, it defines the eventual product in terms which permit it to be understood and controlled by managers, engineers, and/or users who may not be specialists in software technology.
- It constitutes an explicit statement of detailed data processing needs of the system upon which the ensuing computer program design, development, and qualification are based. A significant purpose is to minimize the need for software engineers to further research and interpret system/user requirements.
- It provides a technical basis for developing support documentation of manual and man-machine functions related to operation of the CPCI in the system, e.g., in the form of positional handbooks.
- In defining the allocated baseline, it is the level at which configuration control is maintained over the CPCI throughout the acquisition portions of its life cycle.

It is normal to expect that some information will be missing at the time of initial authentication and approval. Requirements in certain areas are subject to resolution during contract negotiations and firm planning for full-scale development, e.g., with respect to definitions of interfacing equipment characteristics. Various other constraints may also prevent full completion of the Part I specification for a complex mission CPCI in all of its typically massive detail. Rules to be observed in those cases include the following:

- All missing information should be evaluated for its effect on the conduct, cost, and schedule of computer program development. The subsequent preparation of ECPs/SCNs to supply information known to be missing at the time of contract award should be "within scope" of the development contract, and should be scheduled to precede need of the information by computer program designers. All missing definitions of interfaces with other equipment/computer program CIs should be completed prior to PDR.
- Needs to clarify requirements, resolve discrepancies, and add detail to the Part I specification are typical throughout the development process. A continuing function of the developer's system engineering effort is to detect those and correct or expand the specification (via ECP/SCN) whenever indicated. Again, this activity should be a part of the planning, and most of the clarifications should be within scope (see also 4.3.2).

3.4 COMPUTER PROGRAM PRODUCT SPECIFICATION

The computer program product (Part II, or Type C5) specification is basically a comprehensive technical description of the developed computer program CI. As such, it is the principal direct, documentary product of the computer program development effort. Unlike the development specification, it does not have a role as a contractual compliance instrument. Once completed, its primary function is to provide an accurate and complete source of "as built" design data for future use by computer programmers in diagnosing problems and designing changes to the CPCI. It is subject to configuration control following successful completion of its audit at physical configuration audit (PCA), primarily to ensure that it will continue to be maintained in an accurate and current form for those technical uses.

The data item description, DI-E-3120A, is placed on the full-scale development contract primarily to govern delivery of the completed Part II specification, (a) in draft form for review prior to PCA, and (b) in approved form following successful PCA completion. The same DID, modified and so identified by the "/M" suffix, is cited to cover advance delivery of in-process design documentation to be reviewed at PDR and CDR. In the latter cases, however, preparation of the CDRL and backup instructions should observe the following rules:

- The delivery of design documentation is specified in a separate CDRL sequence item for each of the two reviews.
- Each modification statement should specify that format requirements for the product specification set forth in MIL-STD-483 are not mandatory for design documentation reviewed at PDR and CDR.
- The modification statement for the PDR delivery should cite paragraph 30.2.2 of MIL-STD-1521A for required content coverage of the PDR design documentation.
- The modification statement for CDR delivery should require content of the design documentation to include coverage equivalent to all essential content of Section 3 of the product specification with the exception of listings. (Information for other sections to be later provided in the Part II specification format is either not pertinent or not available at the time of CDR.)

Modification of the DID by means of backup instructions is also normally required to govern delivery of the completed product specification. In addition to other "tailoring" to individual CPCIs which may be specified by the procuring activity or proposed by the contractor, the DID itself identifies needs for advance clarification and agreement in two significant areas: (a) the levels of flow charts to be provided in the completed specification; and (b) the specific form in which source and/or assembly listings are to appear.

3.4.1 Content and Development

A completed Part II specification contains descriptive information about the design and coding of the CPI which can be categorized into the following three levels:

- Overall Design. A technical description of the design of the item as a whole, including: identification of computer program components (CPCs), allocations of functions (from the Part I specification) to CPCs, overall design of the CPI data base, storage allocations, timing, sequencing, control logic, and special features.
- Detail Design. A description of each CPC, including: interfaces, limitations, data organization, and such flow charts as are necessary and helpful to understanding the design.

- Listings of the coded computer instructions and data content.*

Basic information for the specification content should be developed incrementally, in parallel with successive stages in design and development of the CPCI. Figure 3-4 depicts an idealized sequence of those stages, the documentation levels, and their relations to technical design reviews and configuration audits. Successive activities shown in the chart are not normally discrete, in the sense that each must be completed before the next begins. Rather, they typically overlap in time, and some of the work performed initially at earlier stages is likely to undergo iteration during one or more of the later steps. Generally, however, the steps should have their beginning and end points in the order indicated. Aspects of the process as a whole which should be understood by configuration managers include those summarized below.

- Technical reviews are accomplished at PDR and CDR on documentation of the in-process design resulting from preliminary and detail design efforts. In each case, the documentation normally serves as an interim "specification"--internally to the developer--to govern the next stage of the overall development process. At those stages, however: that documentation is not formally approved by the procuring activity; it does not function to define contractual requirements; and it remains fully under control of the developer, consistently with his primary contractual responsibility to meet requirements of the Part I specification. As a practical matter, any formal controls external to the technical development activity could seriously impede the continued development, since alterations and refinements during the subsequent steps of analysis, coding, and developmental testing tend to be numerous and frequent.
- Preparation of a completed draft Part II specification is a significant task which should be separately scheduled and accomplished prior to initiating formal qualification testing (FQT) of the CPCI. The task consists basically of: revising and augmenting the existing design documentation as necessary to meet format/content requirements of DI-E-3120A; providing listings in approved form; and verifying all parts of the specification for accuracy, completeness, and understandability in describing the "as built" configuration of the CPCI.

*Listings may be furnished as one or more separate appendices to the body of the specification. However, they are essential and integral parts of the specification for all purposes of identification, control, and specification maintenance.

- The draft of the Part II specification should be available in its initially completed form for inspection at FCA, and should be examined at that time in order to provide guidance to the contractor for his PCA submittal. Configuration control procedures internal to the contractor (i.e., as distinct from technical "baseline management"; see 4.5.1) should be initiated upon completion of the draft Part II and its approval by the contractor's CCB, prior to the conduct of FQIs. Objectives are to maintain control and traceability of all error corrections and/or redesigns which might affect the status of item qualification during the FQI period.

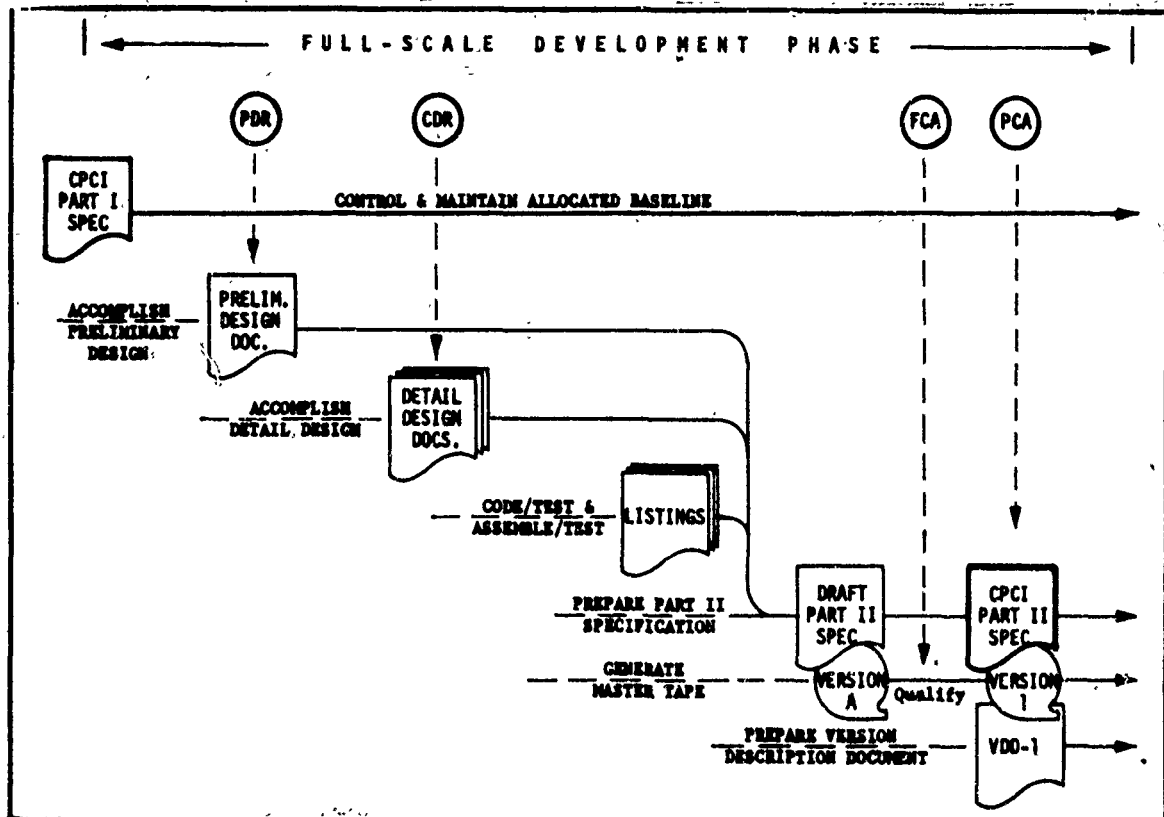


Figure 3-4. Development of the Part II CPCI Specification

3.4.2 Approval and Control

General procedures involved in approval of Type C (Part II) specifications are described in Chapters 2 and 5 of AFSCM/AF LCM 375-7. While formal approval occurs nominally at PCA, it usually entails a number of steps which begin at the time of FCA and may not end until post-PCA actions are completed.

Contractor delivery of the draft Part II for Air Force specification team review should be required not later than 30 days prior to PCA, and should have been preceded by preliminary examination and guidance at FCA (see above). The major objective of the audit as a whole is to verify the specification's adequacy and accuracy as a technical description of the qualified CPCI configuration. In part, that task can be accomplished in a relatively objective manner through comparison of instruction listings contained in the specification with listings generated from the CPCI, at PCA. Verification of descriptive information contained in the specification--i.e., "the prose and flows"--typically requires extensive technical analysis which should be accomplished prior to the PCA data, to the degree permitted by the PO's technical resources.*

PCA should normally be conducted as soon as possible after completing CPCI qualification. However, the latter event may not occur, often, until some time after the pre-PCA delivery data for the draft Part II specification. If test or other changes occur in the CPCI during that draft review period, potential problems in timing of the specification revisions can be resolved by procedures along the following lines:

- Delivery of the CPCI and its first version description document (VDD-1) are timed to coincide with delivery of the draft Part II specification, at least 30 days prior to PCA.
- PCA is conducted on that configuration. Corrections to the draft specification are confined to required improvements in the technical description resulting from the review, not including any changes installed in subsequent test versions of the CPCI.
- The corrected draft is reissued following PCA (e.g., within 15 days) as the authenticated specification defining the initial product baseline configuration of the item.
- Interim changes are processed via ECP and incorporated into the specification through issuance of an initial, post-PCA SCN package to the baselined specification.

*PCA is the event at which the procuring activity formally accepts the CPCI and its Part II specification, as a matter of policy and normal practice. Acceptance is not necessarily total and final, since the DD Form 250 provides for acceptance with shortages. Unaccomplished tests are included as shortages (see para. 5-7,c,(13),(c) of AFSCM/AFLCM 375-7).

The Part II specification continues to function for the remainder of the CPCI life cycle as an "as built" technical description, rather than as a specification (requirements document) in the usual sense. Once baselined initially, it is changed only after coding changes to the CPCI have been designed, developed, and tested--i.e., in effect, fully implemented. This unique characteristic of the computer program Part II specification is an important factor in various aspects of software configuration management. Its relations to special practices in the areas of configuration control and status keeping, and to certain significant discrepancies with established hardware practices, are discussed further in later sections of this guide.

3.5 OTHER SPECIFICATIONS

The Part I and Part II specifications described above normally apply only to computer programs that are custom-developed during a given program, including some which may be developed as significant modifications or expansions to previously-existing computer programs. As indicated earlier, they apply to each developmental item designated as a CPCI, regardless of its size or complexity.

Among the variety of types, subtypes, or forms of specifications identified in MIL-S-83490, MIL-STD-490, and MIL-STD-483, the only ones that apply to computer programs in addition to the Types B5 and C5 are the three listed and described briefly below.

3.5.1 Form 3 Specifications

A Form 3 specification is one specification "form" (as distinguished from "type") defined in MIL-S-83490. Forms are differentiated on the basis of their varying degrees of compliance with the format/content instructions for individual specification types provided in the appendices of MIL-STD-490 (see Table 3-1). That is:

- Form 1a refers to specifications which comply fully with the MIL-STD-490 content instructions, including section/paragraph numbers and titles. The CPCI Part I and Part II specifications described above are normally Form 1a.*

*The use of supplemental instructions in MIL-STD-483, or of modifications via CDRL backup instructions, does not normally affect the Form 1a classification.

- Form 1b permits variations in paragraph numbers and titles, below the section level.
- Form 2 is basically a specification prepared to commercial practice, but complying with supplemental military instructions which are set forth in MIL-S-83490; as written, the Form 2 instructions do not apply to computer programs.

The Form 3 specification is defined as a specification prepared to the contractor's commercial practice, without any military controls. Thus, it is potentially applicable to the procurement of existing "off-the-shelf" computer programs for which the technical documentation is not being developed under the given system contract; and its use for that purpose may be indicated in some cases. However, potential problems exist which should be considered and resolved on a case-by-case basis. As examples:

- Commercial documentation is typically inadequate to perform either the technical or configuration management functions required of specifications for developmental CPCIs. Relations of documentation to actual computer program modules is often such as to prevent ready identification and management of the software assemblies as configuration items. Either (a) planning for computer program support and control should be restricted accordingly, or (b) provisions should be made in the procurement for additional performance and design data to meet the expected needs.
- Questions of data rights should be examined in the light of anticipated needs for duplication and/or maintenance of the documentation, taking into account intended contractor as well as organic responsibilities for the computer program use and support. Special problems may arise if the deployment phase support needs, for example, are not identified until after the contractor to the procuring activity has purchased the software and its documentation from a secondary source.

3.5.2 Inventory Items

If the system can utilize items which are already in Government inventory, such items are identified on an inventory item specification, Type C4. This "specification" consists of a list of the items, together with descriptive material identifying relevant characteristics and applicable documentation. The specification is prepared in accordance with Appendix XII of MIL-STD-490 and supplemental instructions provided in Appendix V of MIL-STD-483.

3.5.3 Addendum Specification

An addendum specification is used to describe the configuration of a new configuration item which is similar to an existing item. Its principal purpose is to reduce the preparation time and bulk of the new specification. Its use is permissible when all of the following conditions are met:

- The new item is a modification of an existing item.
- The existing item is specified fully by a Form 1a specification.
- It is required to retain the existing item and its specification intact, for continuing original purposes.
- There is some reason to establish a relationship between the new and existing items.

The addendum specification is prepared in accordance with instructions in Appendix IV of MIL-STD-483. It consists of a new specification which references the existing specification on a paragraph-by-paragraph basis, noting changes, additions, and deletions. It references a specific issue of the original specification, and from that point on represents a newly-created configuration item separate and distinct from the original. This practice is not often desirable, but has proved useful under some circumstances.

If both the "existing" CPCI and the new CPCI for which an addendum specification is being contemplated are to be developed concurrently for use in the same system, and are to be later controlled by the same deployment phase agency, consideration should also be given to the alternative of classifying the two items as types within a single CPCI (see 2.3 above).

3.6 SUMMARY OF SPECIFICATION ROLES, HARDWARE vs. SOFTWARE

While this guidebook devotes its emphasis to configuration management as it applies specifically to software, needs also exist to draw comparisons in certain areas with hardware practice. In system programs, configuration management of software and hardware are frequently combined, more often than not under the control of personnel whose basic knowledge of the discipline is derived from hardware experience. Specialists in software configuration management are rare; and the military standards frequently fail to clarify how, or whether, requirements in many areas apply to any class of CIs other than

equipment. Hence, this section presents a summary comparison of the two in order to highlight certain fundamental discrepancies in the procurement roles of CI and CPI specifications which account for important differences in configuration management emphasis, phasing, and procedures.

The upper half of Figure 3-5 contains a synoptic diagram of the "model" acquisition process for a weapons system hardware item, together with generalized curves representing a normally-expected distribution of costs over the system's life cycle. This model is chosen for comparison because it (i.e., hardware/ weapons system) represents the acquisition environment in which configuration management evolved, and whose characteristics are reflected throughout the major configuration management concepts and requirements documented in current military standards. Points to be considered in the diagrams, and in comparison with comparable diagrams for software shown in the lower half of the figure, include those summarized below:

- The system specification performs functions in the system program as a whole which are essentially the same for hardware and software CIs. Its primary function is to provide the requirements base from which development (Part I) specifications for CIs and CPIs are derived, and with which they must continue to be related.
- PDR is a comparable event for the two classes of CIs. In both cases, it is an in-process review of CI/assembly-level design, differing appropriately in technical content but not in objectives.
- For hardware, CDR occurs when the CI development as such has been essentially completed. It should normally occur after the completion of sufficient testing, conducted on prototype or R&D articles of the item, to provide reasonable assurance of CI qualification. The primary product of a successful equipment CDR is the decision to release the design to fabrication/production--i.e., in the model case, authorizing the contractor to implement capabilities needed to produce the item in quantity. CDR for software is not a comparable event with respect to either relative phasing or objectives, in that (for example): the development process is still essentially in midstream at the time CPC detail designs are initially completed;* no testing can occur until coding is accomplished; and questions of production costs are normally trivial.

*The term "detail design" as applied to both engineering drawings and CPCs is misleading. The source listings of computer program instructions/data actually represent the level of computer program design which is analogous to detail engineering drawings (cf. MIL-STD-480, para. 4.2.1. vs. MIL-STD-483, para. 140.6.1).

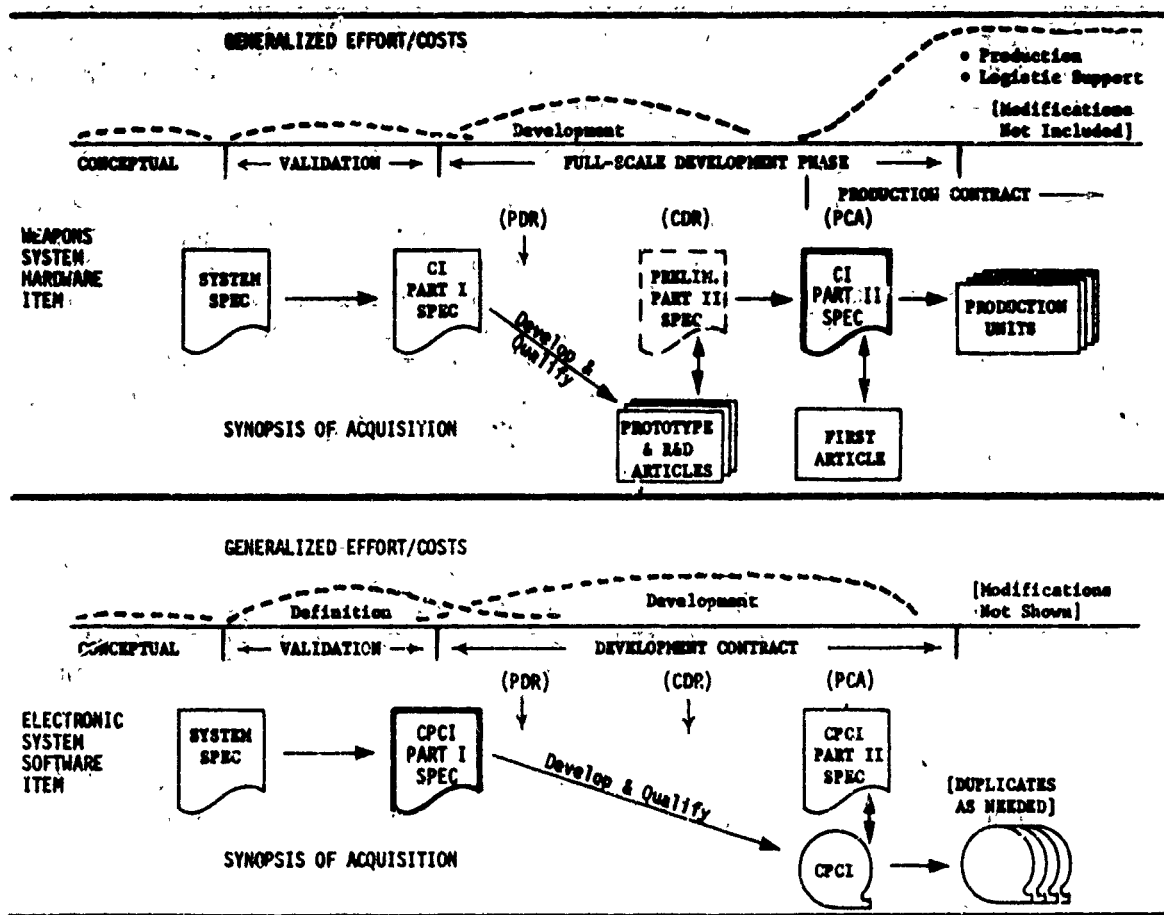


Figure 3-5. Hardware vs. Software - Life-Cycle Factors Significant to Configuration Management.

- PCA is comparable for hardware and software in the sense that it is the event at which procuring activity acceptance of the article and associated documents occurs, and at which the Part II specification is established as the product baseline. Differences in emphasis and procedures stem from significant differences in intended subsequent functions of the Part II specification (see below).

- For hardware, the major focus of configuration management as a whole is typically on control and status accounting procedures which begin when the product baseline is established and expand as the production item is deployed for operational use. This emphasis is consistent with the fact that costs and manpower associated with producing and supporting the item in the field typically account for most of its total costs over the life cycle. In the production contract, the Part II specification serves as the primary contractual instrument and, by virtue of that fact, becomes the direct baseline document against which ECPs are processed. The standard ECP itself reflects the expectation that "total impact" of later changes tends to follow a similar pattern--i.e., costs of development are considered negligible in comparison with impacts on production and logistic support.
- Curves of efforts or costs shown in the diagram are highly generalized. Differences shown in the distributions of effort over full-scale development indicate that the computer program development can normally extend to later in the phase, due to the absence of need for lead time to produce articles required for system DT&E. The principal point of the curves is to illustrate that major equipment costs of production and logistic support are generally absent or negligible for software items, in comparison. The diagram does not attempt to depict generalized costs for modifications. For ground electronic systems, operational phase costs for "software support" are normally significant, but they tend to be predominantly costs of accomplishing modifications.*
- The function of a Part II specification as a technical reference for diagnosing problems and designing modifications is common to both hardware and software. Considering the normal frequency of computer program error corrections and other changes, it represents an essential function which fully justifies formal configuration control at the product level for CPCIs. However, unlike its equipment counterpart: the computer program Part II is not a "build-to", "produce-to", or "test-to" document; if placed on contract, it is a reference as opposed to a compliance document; and, accordingly, it functions in the configuration control process as an impact item rather than as a controlling instrument.

*The situation is slightly overstated, for emphasis. A basic effort is normally required to support the storage, handling, and operation of computer programs, including capabilities to diagnose malfunctions, which can be regarded as over and above the effort of making changes. However, existing regulations have not yet attempted to clarify uniform management and funding distinctions in those areas, for software.

- From the point of view of technical and procurement, as well as configuration management considerations, the Part I specification is the major instrument available to acquisition managers for the control of software, throughout a system life cycle. Thus, as indicated in the diagram, the relative importance of Part I and Part II specifications tends to be the reverse, for software, of that which is normally true for hardware. Implications of this fact are reflected in treatments of configuration control and status keeping procedures described in remaining sections of this guide. Study of the current military standards will reveal that they are also reflected in most of the requirements which have been formulated explicitly for software, but not as yet without some obscurities and inconsistencies.

SECTION 4. CONFIGURATION CONTROL

Configuration control consists of the formal procedures by which changes to system and CI configurations are documented, processed, and authorized. In configuration management, a "change" (or, "engineering change") is really an alteration to the baselined specification which defines the item's required --i.e., approved--configuration. Alterations to a specification being prepared but not yet formally approved and accepted by the procuring activity, or alterations to the physical article itself that do not correspond with changes to the specification, are not changes.

Thus, the procedures of developing and approving specifications described in the preceding section are essential prerequisites to the initiation of configuration control. The control procedures apply only to the system specification during the first part of a system program, but their coverage later expands incrementally as individual CI specifications are completed and successively baselined at the allocated and product levels.

Steps in the control process are relatively simple, in concept. They involve: initiating and documenting a change proposal; reviewing and approving or disapproving the change; and authorizing the implementation of changes that are approved. In working applications, they entail uses of standard forms, organizational roles, and specific procedures which vary in form and complexity as a function of the baseline affected, type or class of configuration item, phase of the program, and other factors.

The guidance in this section is designed to summarize, interrelate, and clarify the application to computer programs of configuration control standards and requirements which are to be found principally in the three sources listed below:

MIL-STD-483 (USAF):

Appendix II	Interface Control
Appendix XIV	Engineering Changes (Computer Programs)

AFSCM/AFLCM 375-7:

Para. 1-12	System Engineering/Design Integration Relationship to Configuration Management
Para. 1-39	Application of MIL-STD-483 (USAF) Appendixes to CPCIs
Chapter 3	Configuration Control

MIL-STD-480:

Basic Standard

Appendix A Instructions for Preparation of ECP

Other sources relevant to individual topics with which the user should also be familiar are:

AFSCP 800-3:

Chapter 9	Configuration Management
Chapter 15	Interface Management
Chapter 20	Program Office Organization

AFR 800-14, Vol. II:

Chapter 6 Configuration Management

4.1 ORGANIZATIONAL FACTORS

Within a program office, activities primarily responsible for matters associated with configuration control are the configuration control board (CCB) and configuration management office (CMO). System prime or associate contractors, and normally their major subcontractors, are required to have the functional counterparts of these activities within their management organizations for the program; names and organizational alignments of the contractor activities may vary, but the functions should be represented.

The program office CCB is the management activity which makes all significant decisions relating to specifications and proposed changes. It is not an organizational unit as such, but a functional body which convenes periodically

and/or on demand. Members consist basically of the chiefs of the PO's organizational units (i.e., engineering, program control, procurement, et al.), plus representatives of the using command, ATC, AFLC, and other organizations involved in the program. The program manager is officially the CCB chairman and bears final responsibility for its decisions; i.e., the membership constitutes an advisory, not a voting, body. Current requirements for CCB membership and operations are described in Chapter 9 of AFSCP 800-3; additional descriptions of actions that can be taken by the CCB on change proposals and use of the CCB Directive (CCBD) for documenting those actions are provided in Chapter 3 of AFSCM/AFLCM 375-7.

The CMO is the center of responsibility within the PO for administrative and staff functions associated with configuration management. Its functions include implementing configuration management policies and procedures, maintaining configuration management files for the program, coordinating and monitoring configuration management actions, processing the review and baselining of specifications, preparing CCB schedules and agendas, and disseminating the results of CCB actions.

Typical relationships of the CCB and CMO to the program office organization are depicted in Figure 4-1. Figure 4-2 illustrates one way in which similar functions may be represented in a contractor's organization for a software development project.

A contractor CCB is chaired by the project manager or his designated representative, and consists of members representing the principal project staff and line activities. Major subcontractors may also be represented on the CCB of a prime or associate contractor. Functions are to approve specifications and change proposals, internally, and to approve the forwarding of proposed actions to the customer CCB. Again, it is a board which meets to issue formal decisions. Those should normally be based on recommendations of the individual members derived from their study and coordination of each agenda item in advance of the meeting.

Functions of the contractor's CMO should be generally similar to those of the program office CMO. The contractor CMO is responsible for:

- The contractor's configuration management plan, which should be prepared or updated and approved early in the full-scale development phase.
- The preparation and control of documented internal standards/procedures for configuration management, covering events and processes affecting all organizational units of the project.

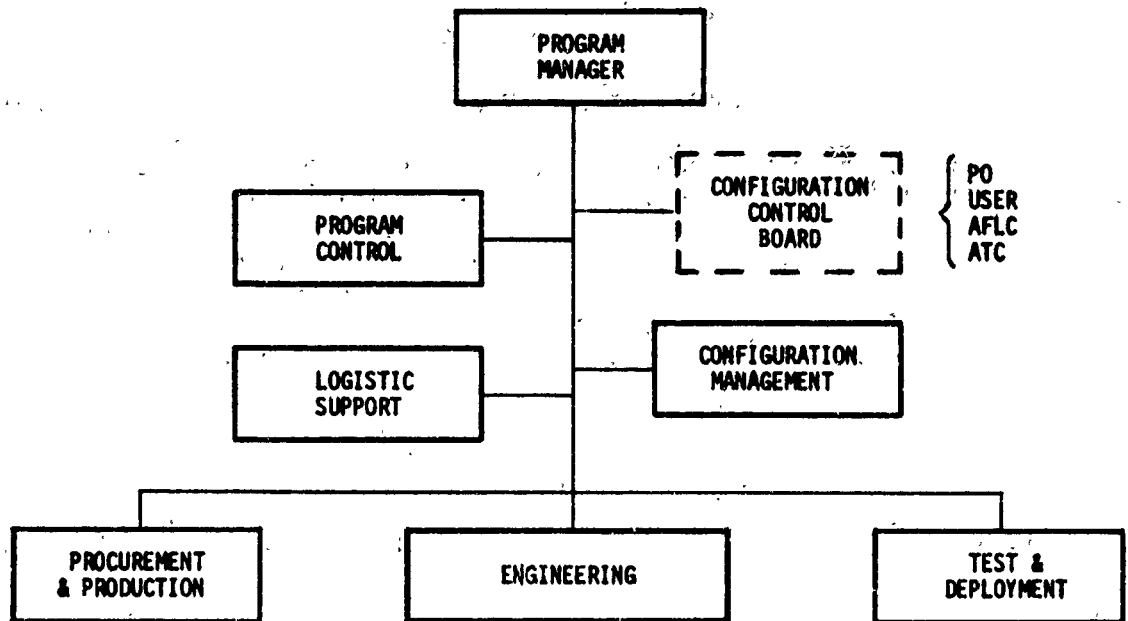


Figure 4-1. CCB and CMO Relationships to a Program Office

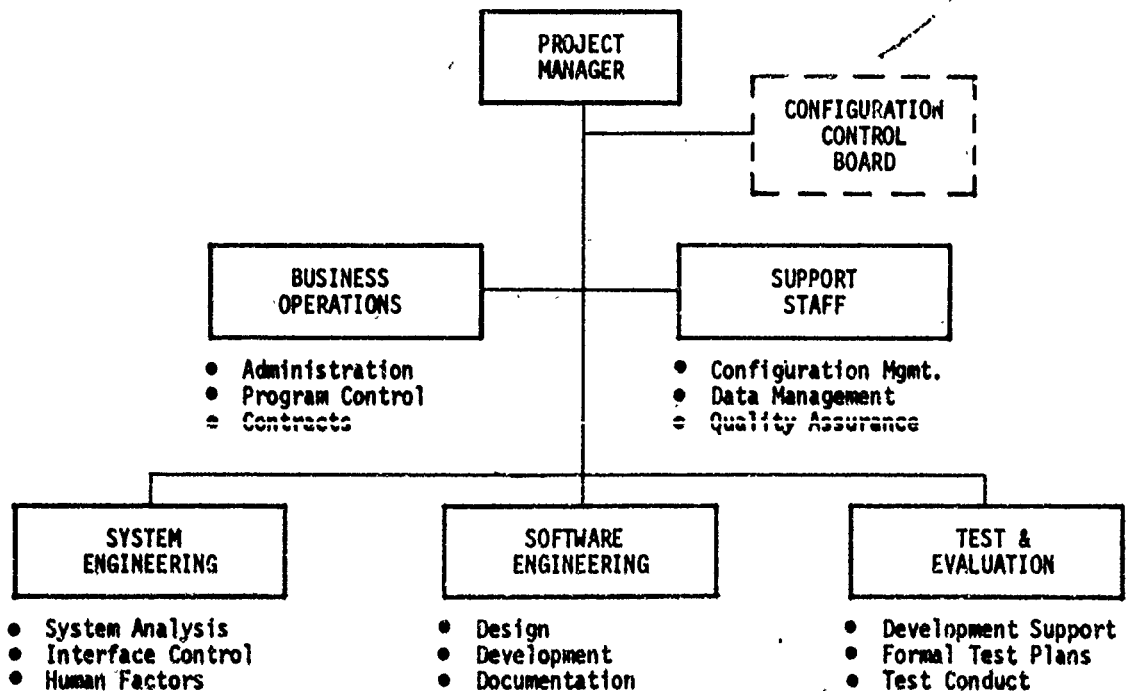


Figure 4-2. Illustrative Organization for a Software Contractor

- Control of change processing and related internal configuration management actions; liaison with the program office CMO; and serving as secretariat to the contractor CCB.
- Collecting and maintaining status data, in coordination with specification/document submittals and change processing events, and issuing periodic reports of documentation and change status.

Documented internal procedures should be "tailored" to the individual project and contractor's project organization, to the level that requirements and responsibilities are clearly delineated in relation to individual technical and staff activities. Several specific topics for such standards are suggested in other paragraphs of this guidebook, including certain areas in which configuration management procedures must be closely coordinated with those in quality assurance and data management, in particular.

4.2 CHANGE CLASSIFICATION

All changes to established baselines are distinguished for purposes of change processing and control as being either Class I or Class II. In general, Class I are the more important changes, which must be formally proposed by the contractor and approved by the procuring activity CCB prior to being implemented. Class II are the relatively minor changes which can be implemented by the responsible contractor without prior approval, but which must be reported for procuring activity review and concurrence with their classification.

Formal definitions of the factors determining classification of computer program changes are provided in MIL-STD-483, paragraph 140.6. In essence, they are as follows:

- A change must be classified as Class I if it affects a technical requirement contained in the Part I specification, the contract schedule, or costs. Changes to the Part II specification are also Class I if they affect the design (excluding listings), and whenever they affect CPCI performance or external interfaces--i.e., whether or not the latter are actually specified in the Part I specification, as they should be.
- All changes which do not meet the definitions of Class I changes are Class II. Examples are editorial changes to correct specification errors, or to clarify expressions, and changes in the computer instruction/data listings (in the Part II specification) to reflect corrections of computer program errors.

Questions often arise regarding interpretation of the criteria with respect to "editorial" corrections, e.g., in clarifying conflicting or ambiguous statements of requirements, and with respect to the meaning or permissible magnitude of computer program "errors". It is usually necessary to arrive at working interpretations and establish more specific rules for borderline cases through procuring activity/contractor coordination and agreement in each project.

It is a frequent misconception that the difference between Class I and Class II is really a matter of "cost" vs. "no-cost" changes. While it is true that Class II changes should always be "no-cost"--i.e., impact on costs established in the contract--the reverse is not true for Class I changes. Compatibility changes, for example, must be within the scope of existing contract requirements by their MIL-STD-480 definition. For computer programs, Class I changes which expand and refine the requirements of Part I specifications prior to qualification testing are to be encouraged (cf. 3.3.3 above).

4.3 CLASS I CHANGE PROCESSING

The treatment of configuration control in this section emphasizes control during the full-scale development phase of a system program. That phase is assumed to extend beyond the point of PCA for developmental CPCIs to include a period towards the end of the phase (e.g., through system DT&E) during which the original developer is responsible for proposing and implementing changes at both allocated and product baseline levels.

The full controls in effect at the end of that period are capable of being extended indefinitely without further expansion of the procedures. However, organizational responsibilities for both controlling and implementing changes during the deployment phase will shift at program management responsibility transfer (PMRT) from the PO and original developer, respectively, to (a) the supporting command (normally AFLC) and (b) an in-house computer programming support group and/or other contractor(s).*

Since PCA normally occurs for CPCIs towards the end of full-scale development (usually, just prior to the conduct of system DT&E), the bulk of change processing activity during most of the phase as a whole occurs at the allocated baseline only. Although control expands to include the Part II specification

*Configuration control and engineering responsibility for each system as a whole are transferred to AFLC. Depending on agreements reached for each system and documented in the CRISP and O/S CMP, control of mission CPCIs may transfer to a using command computer program configuration sub-board (CPCSB; see Chapter 6 of AFR 800-14, Vol. II).

at PCA, Class I changes beyond that point continue to be addressed primarily to the Part I for reasons outlined in the preceding section (see 3.6). Briefly: (a) the Part II is a description of the end product, not a requirements document; and (b) changes to the design of a qualified CPCI which do not result from changes in required performance should not normally be permitted. Changes in the Part II specification listings to track corrections of computer program errors (i.e., in essence, failures of the CPCI to fully qualify) should normally be Class II.

It is an important factor in control actions during full-scale development, however, that the technical impact of a given Part I specification change tends to expand progressively from the outset of the phase, up to and including PCA. A change which may affect only the Part I specification itself, initially, will later cause redevelopment of the affected computer program elements to the extent that successive stages of the overall design/development/test sequence have been completed. It is also of concern to configuration managers responsible for tracking the implementation of approved changes that other maintainable documents enter the process as they are delivered and approved during the phase, including test documents, handbooks or manuals, and the version description document as well as the CPCI and its Part II specification.

4.3.1 Two-Step Processing

Change processing actions consist largely of handling information which is contained on or with two standard forms known as the engineering change proposal (ECP) and specification change notice (SCN).

Standard format for the ECP is prescribed and illustrated in MIL-STD-480. The form consists of six separate pages, designated as DD Forms 1692 through 1692-5. Although designed basically for proposed changes to equipment, it is also the only existing form which is approved for use by contractors in proposing changes to the system or software specifications. Instructions for appropriate modification and use of the form are provided, however: (a) in MIL-STD-480 and MIL-STD-483, Appendix XIII, for the system specification; and (b) in Appendix XIV of MIL-STD-483 for computer program changes. In the latter case, only the first two pages of the form (i.e., DD Forms 1692 and 1692-1) are used.

Standard format and instructions for preparation of the SCN are provided in MIL-STD-490. The SCN is normally used as a cover sheet to a set of specification change pages containing exact changes to the affected paragraphs. Format and uses of the SCN in relation to procedures of computer program document maintenance are discussed further in the next section. Roles of the SCN in processing ECPs are amplified below.

In the traditional model of change processing derived from equipment practice, the two forms are prepared, submitted, reviewed, and approved or disapproved together as parts of a single "ECP package", which consists principally of the formal ECP plus one SCN for each affected specification. Approval of the proposed change by the procuring activity CCB results in incorporating the specification revisions into the contract, thus authorizing the contractor to alter his further development or production of the item in accordance with the new requirements. The established assumption is that the cost of developing the specification changes as such is negligible in comparison with the subsequent costs of implementing the change--which it typically is, in the equipment environment.

"Two-step" processing of Class I changes to computer programs refers to the practice of submitting a given ECP in two sequential steps, first as a formal ECP which is not accompanied by the SCN to an affected specification and subsequently as a revised ECP to accompany the completed SCN. Procuring activity approval also occurs in two steps, in that: (a) approval of the formal ECP results in authorizing the contractor to expend the effort required to develop the specification revisions; and (b) approval of the revised ECP is contingent upon approval of the completed SCN.

General requirements pertaining to two-step processing are stated in paragraph 140.6.3 of MIL-STD-483. The intent of the procedures is to recognize that development of the SCN itself can be an important portion of the total cost of implementing some computer program changes. The rules for relating SCNs for different specifications to ECPs are summarized below to illustrate how the procedure should apply in accordance with that intent.*

- System Specification. A proposed change to the computer program Part I specification may necessitate a change to the system specification. In that case, the formal ECP must always be accompanied by an SCN to the system specification at the time of its initial submittal.
- Part I Specification - Minor Changes. SCNs covering proposed change pages to the Part I specification should accompany ECPs prepared to accomplish expansions or refinements (i.e., eliminating "TBDs" or other areas of inadequacy within the original intent; cf. 3.3.3 above). SCNs should also accompany submittal of the formal ECP at all other times when the information is needed in that form to support CCB decision and when cost of their preparation is not substantial.

*The "SCN" as discussed here refers to the cover of a complete package of change pages to the specification, in a form suitable for distribution to update the specification. The nature of the change, and identified effects of the change on parts of each specification, must be described in the formal ECP itself, whether or not accompanied by the SCN.

- Part I Specification - Major Changes. Two-step processing applies to the Part I when the preparation of the specification change pages represents a significant portion of the total effort of implementing the change, and when the nature of the change can be described adequately to support CCB decision in the ECP itself. Examples are the addition or deletion of significant required capabilities of the CPCI, which may entail extensive system engineering analysis and result in changes to many pages of the specification.
- Part II Specification. SCNs to the Part II do not occur until after PCA. Although they should normally result from ECPs addressing both parts, the possibility does exist that ECPs may be processed against the Part II only. In either case, the formal ECP is not accompanied by an SCN when initially submitted. Two-step submittal always applies, since the completion of changes to the Part II specification (as built) represents the end-point of implementing any computer program change.

Thus, two-step processing may apply to the Part I specification alone at any time during full-scale development prior to PCA. It may also apply to the Part I after PCA, depending on the given change, and it always applies to the Part II. Figure 4-3 illustrates the general sequence and elements of the process for the "maximum" case of a change (a) which affects everything related to the CPCI, and (b) for which implementation is to be completed after PCA.* The diagram is highly simplified with respect to certain factors mentioned in the following comments:

- In this example, the formal ECP is not accompanied by an SCN to either part of the CPCI specification. It must be accompanied at the outset by an SCN to the system or a system segment specification, however, if one of those is affected.
- The diagram of a two-step change completed prior to PCA would eliminate the middle band of events (i.e., "middle" from top to bottom) as a visible part of the change activity, together with those impact documents represented in the lower band that are not yet delivered. Typically, the

*Those can include some changes which were actually initiated well in advance of PCA. As the PCA date approaches, schedules for ECP implementation must be examined and adjusted to avoid conflict with the pre-PCA period required for draft Part II specification review. A "cutoff" date may have to be established prior to the draft Part II delivery, such that changes to be implemented after that date are nominally processed as post-PCA changes. See 3.4.2 above.

CPCI test plan and procedures are affected by, and maintained to reflect, all such Part I-only changes so that their presence in the initial version of the CPCI can be verified during qualification tests.

- Class I changes affecting the Part II specification only are possible. In those cases, events shown for the Part I specification, and for unaffected impact documents, are naturally eliminated.
- A second (revised) ECP is prepared to accompany delivery of SCNs to the CPCI specification. In the usual case, the SCNs and other products shown at the far right in the diagram are likely to be completed and delivered for review and approval over some distributed period of time, rather than simultaneously.
- As this diagram may suggest, the computer program change process as a whole tends to constitute a repetition of the original, total development cycle, in greater or less degree depending on magnitude of the change.

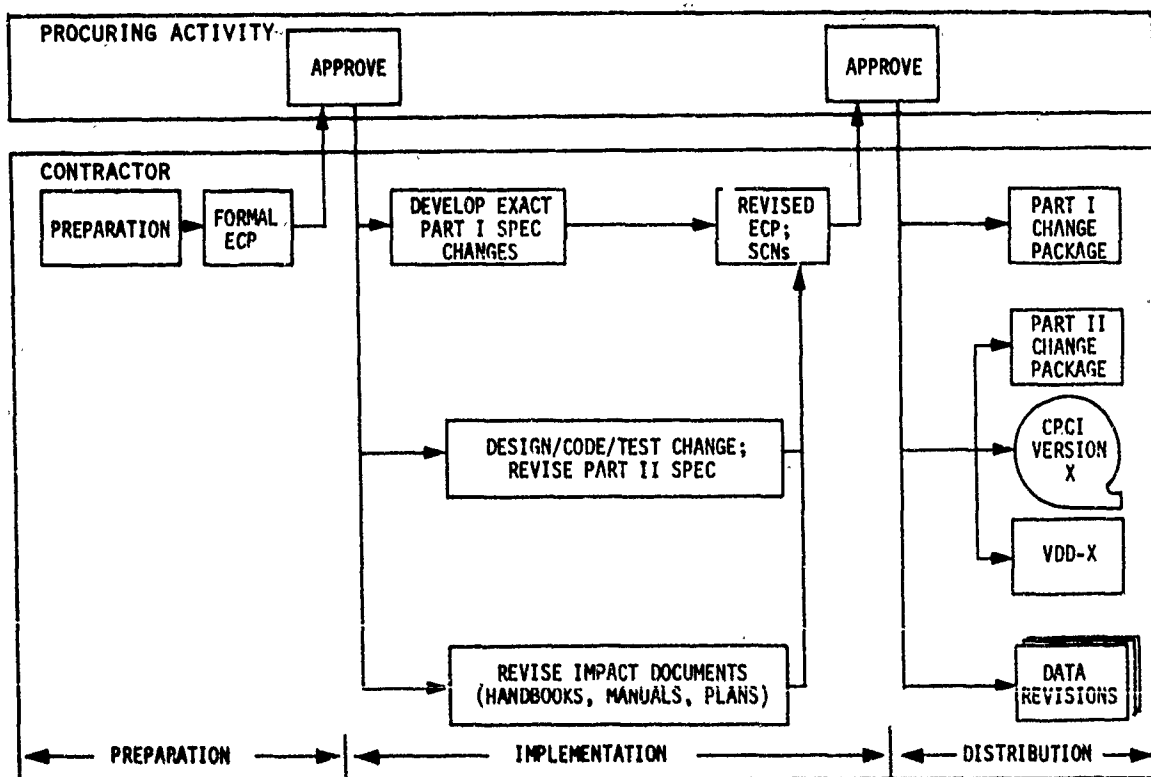


Figure 4-3. Synoptic Diagram of Two-Step Processing. The diagram illustrates the case of an ECP which affects all delivered items, following PCA. Typical differences in relative timing of SCNs and other change impact products are not shown.

4.3.2 Preparation of ECPs

It is an underlying premise at the time of a system contract award that the contractor will perform services and deliver products exactly as specified (i.e., "nothing less and nothing more"). In practice, events always occur during the period of an extended development cycle to alter the procuring activity's requirements, or the contractor's ability to meet the original requirements, or some combination of those. From that point of view, configuration control provides a mechanism to deal flexibly with those events as they occur and at the same time to preserve the spirit of the basic premise. Applied in proper coordination with engineering and other support management functions (notably, program control and procurement), the controls permit contractor performance to be judged against contractually-specified technical requirements, schedules, and costs which are kept up to date throughout the development period.

The need for a change to the approved configuration of a given CPCI may be identified originally by sources within the Air Force, by the responsible contractor, or by other contractors. Whatever the original source, however, an essential first step in the change processing cycle is the preparation of a formal ECP by the responsible contractor. Figure 4-4 illustrates the two pages of the standard ECP form used for that purpose. Blocks crossed out on Page 1 are "not applicable" to computer programs. Other blocks are to be completed in accordance with instructions provided jointly in MIL-STD-480 and MIL-STD-483, using continuation sheets attached to the standard form when additional space is needed. The general nature of information called for in the body of the form is summarized briefly as follows:

- Identification of affected specifications, including the computer program functions, CPCs, and specification paragraphs affected by the proposed change.
- A description of the change, to a level of detail adequate for CCB decision, referencing the SCR(s) when provided with the ECP.
- A justification for the change, in terms of the problem to be resolved or new capability to be provided, referencing directives or other supporting documents.
- A summary of alternative solutions considered, referencing trade studies and reports.

COMPUTER PROGRAM									
ENGINEERING CHANGE PROPOSAL, PAGE 1 (SEE HEC-STD-140 FOR INSTRUCTIONS)					ENGINEERING CHANGE PROPOSAL, PAGE 2 (SEE HEC-STD-140 FOR INSTRUCTIONS)				
1. ORIGINATOR NAME AND ADDRESS					2. CLASS OF ECP				
3. ECP DESIGNATION					4. PRIORITY				
5. SPECIFICATIONS AFFECTED					6. DRAFTING AFFECTED				
7. TEST PLAN					8. DRAFTING AFFECTED				
9. TITLE OF CHANGE					10. CONTRACT NO. & LINE ITEM				
11. CONFIGURATION ITEM IDENTIFICATION					12. PRODUCTION				
13. NAME OF PART OR LOWEST ASSEMBLY AFFECTED					14. PART NO. OR TYPE DESIGNATION				
15. DESCRIPTION OF CHANGE					16. EFFECTS ON CONFIGURATION ITEM SPECIFICATIONS				
17. EFFECTS ON PERFORMANCE, INTEGRATED LOGISTIC SUPPORT, MAINTENANCE, OPERATIONAL EFFECTIVENESS, ETC.					18. EFFECTS ON CONFIGURATION ITEM SPECIFICATIONS				
19. DEVELOPMENTAL REQUIREMENTS AND STATUS					20. EFFECTS ON CONFIGURATION ITEM SPECIFICATIONS				
21. TRADE OFFS AND ALTERNATIVE SOLUTIONS					22. ESTIMATED NET TOTAL COSTS				
23. SUBMITTING ACTIVITY AUTHORIZING SIGNATURE					24. APPROVAL				
25. CLASS 1					26. CLASS 11				
27. DATE OF ACTION					28. DATE OF ACTION				
1. ORIGINATOR NAME AND ADDRESS 2. CLASS OF ECP 3. ECP DESIGNATION 4. PRIORITY 5. SPECIFICATIONS AFFECTED 6. DRAFTING AFFECTED 7. TEST PLAN 9. TITLE OF CHANGE 11. CONFIGURATION ITEM IDENTIFICATION 13. NAME OF PART OR LOWEST ASSEMBLY AFFECTED 15. DESCRIPTION OF CHANGE 17. EFFECTS ON PERFORMANCE, INTEGRATED LOGISTIC SUPPORT, MAINTENANCE, OPERATIONAL EFFECTIVENESS, ETC. 19. DEVELOPMENTAL REQUIREMENTS AND STATUS 21. TRADE OFFS AND ALTERNATIVE SOLUTIONS 23. SUBMITTING ACTIVITY AUTHORIZING SIGNATURE 25. CLASS 1 27. DATE OF ACTION 12. PRODUCTION 14. PART NO. OR TYPE DESIGNATION 16. EFFECTS ON CONFIGURATION ITEM SPECIFICATIONS 18. EFFECTS ON CONFIGURATION ITEM SPECIFICATIONS 20. EFFECTS ON CONFIGURATION ITEM SPECIFICATIONS 22. ESTIMATED NET TOTAL COSTS 24. APPROVAL 26. CLASS 11 28. DATE OF ACTION					1. ORIGINATOR NAME AND ADDRESS 2. CLASS OF ECP 3. ECP DESIGNATION 4. PRIORITY 5. SPECIFICATIONS AFFECTED 6. DRAFTING AFFECTED 7. TEST PLAN 9. TITLE OF CHANGE 11. CONFIGURATION ITEM IDENTIFICATION 13. NAME OF PART OR LOWEST ASSEMBLY AFFECTED 15. DESCRIPTION OF CHANGE 17. EFFECTS ON PERFORMANCE, INTEGRATED LOGISTIC SUPPORT, MAINTENANCE, OPERATIONAL EFFECTIVENESS, ETC. 19. DEVELOPMENTAL REQUIREMENTS AND STATUS 21. TRADE OFFS AND ALTERNATIVE SOLUTIONS 23. SUBMITTING ACTIVITY AUTHORIZING SIGNATURE 25. CLASS 1 27. DATE OF ACTION 12. PRODUCTION 14. PART NO. OR TYPE DESIGNATION 16. EFFECTS ON CONFIGURATION ITEM SPECIFICATIONS 18. EFFECTS ON CONFIGURATION ITEM SPECIFICATIONS 20. EFFECTS ON CONFIGURATION ITEM SPECIFICATIONS 22. ESTIMATED NET TOTAL COSTS 24. APPROVAL 26. CLASS 11 28. DATE OF ACTION				

DD FORM 1692-1

DD FORM 1692

Figure 4-4. The Computer Program ECP Form

- Identification of required tasks and schedules for accomplishing, as applicable to the given change: (a) analysis and preparation of changes to the Part I specification; (b) redesign, coding and testing of changes to the CPCI; (c) preparation of Part II specification changes and a new version description document; and (d) revisions of other maintainable documents impacted by the change.
- Identification of impacts of the proposed change on other systems or configuration items, and on personnel or other factors affecting the system program.
- A dollar estimate of the effect on contract costs if the change is approved.

Detailed instructions for most of the information indicated above are provided in Appendix XIV of MIL-STD-483. Those instructions are written in the form of a supplement to MIL-STD-480, however--i.e., requiring the user to consult the latter for instructions and related general rules which are not specifically modified or replaced in MIL-STD-483. Because of the variable interpretations that can be made of that distributed source material, in addition to its awkward arrangement for software users, this is a topic (among others; cf. 4.1 above) which the contractor's configuration manager should address and consolidate into one, self-contained internal procedures directive for uniform use by project personnel responsible for preparing ECPs.

Further expansion and tailoring of the source instructions is needed for software applications in general as well as for each project. As examples, rules in the following areas should be examined, clarified; and applied based on coordination with (and, where indicated, direction by) the procuring activity CMO:

- ECP Justification Codes. Policies for the use of justification codes in the given program should be established by the program office CMO and provided to the contractor. In general, software changes are confined to those in the first two categories listed in paragraph 4.3 of MIL-STD-480, i.e., correction of deficiency and operational/logistic support. Cost reduction changes are conceivable, but rare. Production stoppage does not apply, except that the separate record-only code applies to all ECPs when so indicated by the contracting method.
- ECP Types. Preliminary (Type P) ECPs apply to computer programs in the manner stated in MIL-STD-480. In addition, use of a revised type (Type R) is recommended for those revisions which are issued to accompany the submittal of SCNs authorized by previously-approved formal ECPs when two-step processing applies. Such a revised type of ECP also carries the normal designation of a revision as required in Block 5f of the ECP form.

- Related ECPs. Related engineering changes occur when the proposed change to one CI (the basic change) requires changes to other items for purposes of compatibility. In those cases, a separate ECP is prepared for each affected CI and cross-references are made in or with all of the ECPs to identify the relationship, whether within or across contractors. Requirements set forth in paragraphs 4.8.3 and 4.8.4 of MIL-STD-480 apply to computer program and equipment CIs, both jointly and separately, although the basic ECP is not often addressed to a computer program item when both are involved. In this area, one particular need is to clarify coordination requirements across contractors for purposes of related ECP status reporting (see 5.3).

Internal directives prepared by the contractor's CMO should cover organizational responsibilities and procedures, as well as content requirements, for ECP preparation. Examples of internal preparation procedures are illustrated in Figures 4-5 and 4-6, using as a model the contractor project organization outlined previously in Figure 4-2. The examples are chosen to illustrate how the preparation might occur (a) for within-scope changes to the Part I specification which are, in effect, completely implemented at the time of ECP submittal, and (b) the more complex changes for which significant further implementation effort depends on procuring activity CCB approval of the ECP. The two examples also tend to be typical of "no-cost" vs. "cost" changes, although that distinction does not necessarily hold in all cases.*

In the first example (Figure 4-5), the typical circumstance is when a Class I change is being prepared to add previously missing or incomplete information, e.g., eliminating "TBDs" for detailed definitions of certain inputs, outputs, processing requirements, or external interfaces. Completion of the SCN to the Part I specification, through system engineering effort previously budgeted for the purpose, is the event which triggers preparation of the ECP. Hence, in this case: the SCN accompanies the ECP; the change is completely implemented when the SCN is approved and distributed; and, by virtue of the latter fact, the ECP entails no estimation of costs. It should generally be true of such changes that they do not alter requirements in ways which make it necessary to undo and repeat steps already taken in computer program design and development; rather, they supply details and clarifications which support the development process.

*"Cost" vs. "no-cost" is distinguished specifically by the presence or absence of a dollar amount in Block 21 of the formal ECP, identifying estimated effects on contract cost if the proposed change is approved.

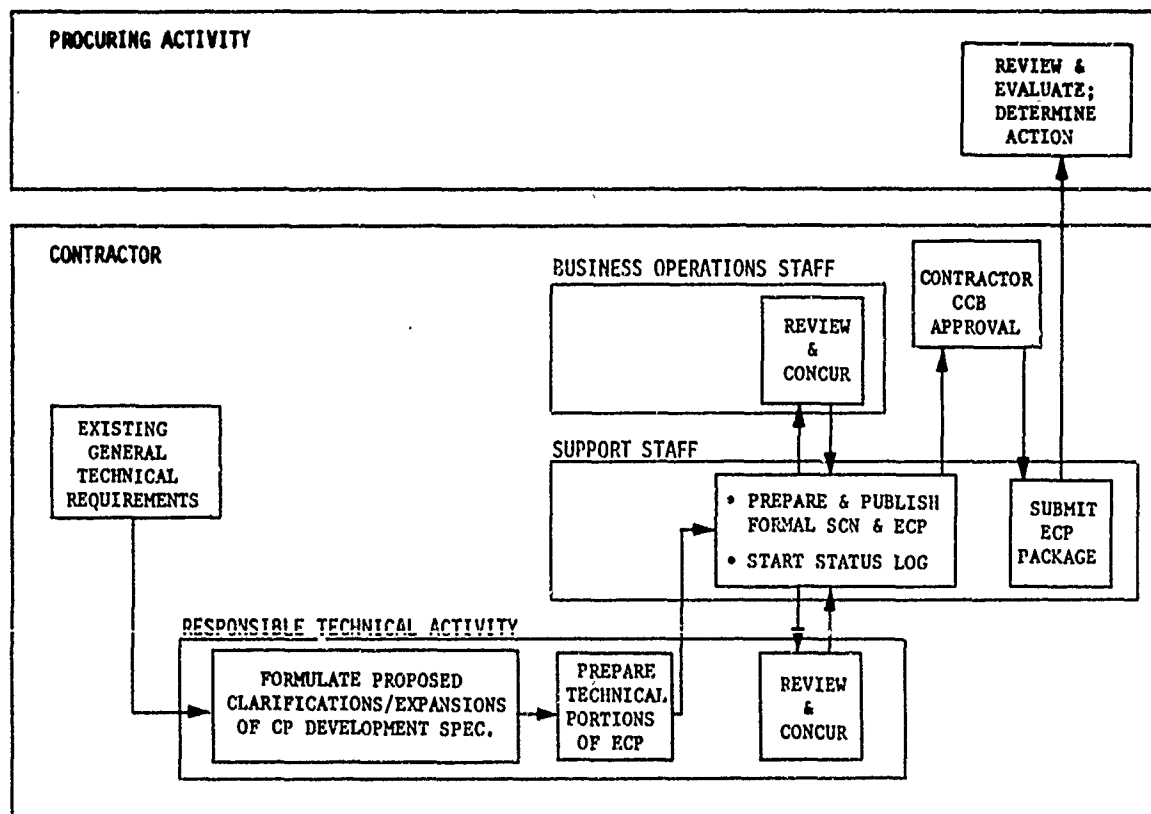


Figure 4-5. Initiation and Preparation of ECPs Proposing Within-Scope Expansions of the CPCI Part I Specification (see text).

Procedures illustrated in the second example (Figure 4-6) apply to Class I changes which add to or alter previously-defined requirements in the baselined specification and call for contractor implementing efforts to be initiated, or not, as a result of actions taken by the program office CCB. In C³ systems, such changes affecting the mission CPCIs stem from various sources and, in the aggregate, tend to be numerous.. They include changes to the CPCI configuration resulting from system specification changes to accommodate new or revised interfaces with other systems, changing operational requirements of the using command, and other needs for capabilities not covered in the initial program. They may also include changes for which needs are identified initially by the contractor or other participants as a result of analysis and testing accomplished during the program.*

*This guide does not attempt to address the many contingencies which can arise when the Part I specification is missing or grossly inadequate, although such cases are all too frequent. The standards do not provide for orderly configuration control, nor for acquisition management of computer programs in many related areas, under those circumstances.

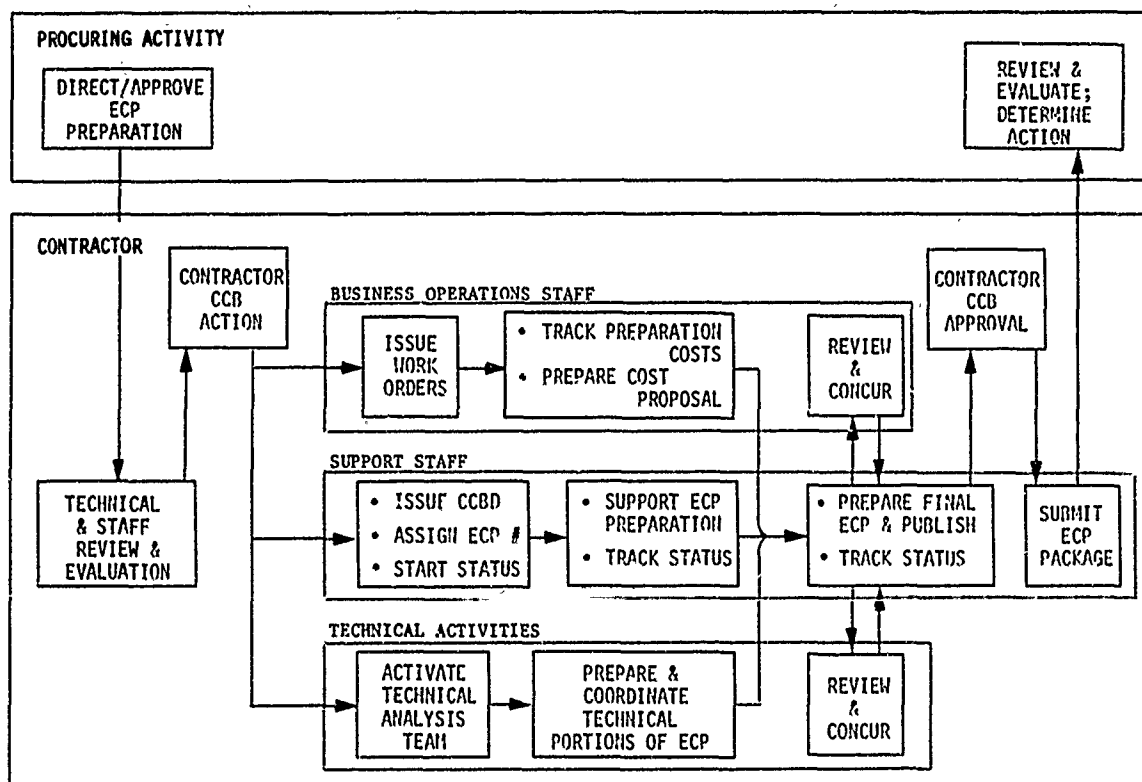


Figure 4-6. Initiation and Preparation of a Class I Change Proposal Involving Subsequent Implementation Effort (see text).

The preceding Figure 4-6 indicates that the preparation process is initiated by PO direction, which should normally be true whether the need is identified originally in-house or by the contractor. If originated by the contractor, the period shown would have been preceded by a preliminary ECP (Type P) and/or other advance coordination leading to the PO direction. This diagram as a whole represents an expansion of the "Preparation" portion of the earlier Figure 4-3, illustrating two-step processing. SCNs will accompany the ECP or be submitted later in accordance with the rules summarized for two-step processing in 4.3.1 above.

4.3.3 Program Office CCB Actions

In its role as secretariat to the CCB, the program office CMO receives and processes ECP packages submitted by contractors. The ECPs are scheduled for review in accordance with formal agendas prepared and distributed by the CMO in advance of CCB meetings. The CMO also initiates and maintains a status log

or report for each ECP, which begins with the date of receipt from the contractor and continues until all suspense dates associated with the ECP have been satisfied by appropriate action.

Formal review of each ECP by the CCB results in one of the following four actions:

- a. The ECP is approved as written.
- b. The ECP is disapproved.
- c. The ECP is approved with specific changes.
- d. Action is deferred, either for further investigation as directed by the CCB or for resolution by higher headquarters.

The action taken is recorded, together with other information relating to the ECP, on a CCB Directive (CCBD) prepared by the CMO for signature by the CCB chairman. The CCBD itself receives in-house distribution only, as the document which provides direction to elements of the program office regarding further actions to be taken on the given ECP. It includes specific requirements to be observed by the contracting officer in preparing and issuing notification of contractual coverage of the ECP to the contractor.

4.4 CLASS II CHANGES

It was indicated earlier that the changes dealt with in configuration management consist most directly of alterations to baselined specifications. That principle is true for all changes to computer programs, whether classified as Class I or Class II. The difference between the two classes is a matter of established definitions relating to the importance of the change, such that Class II changes are those which do not really alter the intent and scope of technical requirements, or impact contract schedule or costs. Hence, they are changes which can be accomplished by the contractor without asking advance approval by the procuring activity.

However, each Class II change has to be reported for information and concurrence with its classification. Non-concurrence can result in direction to remove the change and to reclassify it as a change subject to Class I processing and approval before being restored.

Class II changes can be reported on the standard ECP form (using only Page 1) or on a contractor's own form. In the latter case, the form must contain minimum information specified in MIL-STD-480 (paragraph 4.6.2), consisting of identification of the affected item and part, description of the change, justification, and contract number. Forms similar to that illustrated in Figure 4-7 have often been used for reporting Class II changes to computer programs. In this guidebook, that document is referred to as a "Class II Change Report (CR)" rather than as a "Class II ECP", since it is in fact a report, not a proposal. Use of the "CR" designation also permits ready distinction with ECPs (always Class I, herein) when the two types of document are listed together in status reports.

Requirements pertaining to Class II changes contained in the configuration management standards tend to be scattered and limited, particularly for computer programs. As in other areas, this is a topic which merits specific coverage in the contractor's configuration management plan and internal procedures, based on clear understanding and approval by the program office CMO. The following list outlines the nature of policies and procedures to be considered and clarified for application in each program, taking into account necessary relations of Class II change processing with other aspects of software configuration management.

- Each Class II CR is addressed to either the Part I or the Part II of a CPCI specification, but never to both. A change which affects any other delivered, maintainable document must be proposed and processed as a Class I change. In general, the total impact of a CR must be confined to the given Part I or Part II specification addressed.
- SCNs to the affected specification are not normally issued for the sole purpose of incorporating Class II changes. As a rule, Class II changes are included in SCNs issued to incorporate Class I changes, and a separate CR is also included with the ECP package to identify each Class II change accomplished since the preceding issue of an SCN to that specification or volume. Thus, a given ECP package may consist of one ECP, some number of SCNs (see 5.1.3), and some additional number of CRs at the time of its submittal.
- CRs, as well as the ECP, are identified by numbers and titles on each SCN affected. Thus, after approved SCNs are distributed and inserted into copies of the specification, each copy contains a record of both Class I and Class II changes incorporated in the given issue.

COMPUTER PROGRAM CLASS II CHANGE REPORT				
ORIGINATOR			DATE	
SPEC NO./PART	CONTRACT NO.	CR NO.	REV.	CORR.
CPCI NOMENCLATURE				
TITLE OF CHANGE				
DESCRIPTION OF CHANGE				
JUSTIFICATION				
RELEASED BY		AUTHOR		
CLASSIFICATION APPROVAL			DATE	

Figure 4-7. Sample Contractor's Form for the Class II Change Report (CR).

- A continuing record of all CRs issued against the Part I specification is included in Section I of the computer program configuration index (see 5.2), in the form of a listing which identifies each CR by number and title, together with number and issue date of the affected SCN. Following PCA, a similar record is maintained for all CRs to the Part II specification, in Section II of the index.
- Class II changes to the baselined Part II specification include, as one prominent subclass, changes to the listings to reflect computer program error corrections. The version description document issued to accompany each version or interim version of the CPCI (following the initial version; see 5.4.3) must identify all such Class II changes installed in the CPCI

since the preceding version, by CR number, title, and issue date. A continuing record of these CRs is also maintained for all past issues of the version description document in Section VI of the configuration index.*

4.5 RELATED CONTROLS

This section addresses topics related to configuration control which have proved to be subjects of frequent questions and occasional misconceptions. In this guidebook, as in the military standards, the treatment of software configuration management emphasizes formal controls and tasks in which configuration managers and centralized CCBs are directly responsible. Attention is focused on the completion, control, and status of baselined specifications. Some of the questions relate to the absence of procedures for controlling design documents, listings, and tapes or card decks. Others relate to the absence of requirements in certain areas which are familiar in hardware configuration management. As suggested in the comments below, the reasons for the missing coverage in the standards (and elsewhere in this guide) are varied.

4.5.1 Baseline Management as a Technical Tool

The general point has been made in preceding sections that configuration management expands in discrete steps as specifications are completed and baselined successively at the functional, allocated, and product levels. Prior to each of those steps, however, the technical documentation which leads to the completed specification typically evolves through many levels, forms, and iterations during the course of its development. In situations where the given system or CI specification development requires many analysts/designers, working concurrently on separate portions of the total task, some engineering managers have adopted the generalized techniques of baseline management as their own set of tools for exercising systematic control over that process.

*In practice, the process of error analysis, correction, installation, and testing occurs first in the CPCI. The Part II specification update occurs "after the fact" to record those corrections judged to be acceptable. Although many such corrections to the code may be small, systematic measures to assure that they are controlled and recorded are essential, since a loss of visibility at that level can easily result in the familiar phenomenon of a CPCI gradually losing any known relationship, over time, with its specification.

As applied in that framework, specifically: the initially approved design at each level is documented; each proposed expansion, refinement, or other alteration is examined for its impact; the working documents are altered to reflect all approved refinements; the current status of approved design is made known to all affected participants; and records may be kept to provide an "audit trail" as the design evolves. The alterations are likely to occur on an active and continuing basis as design information is developed and added at successively more detailed levels. Thus, the concept of a "progressively expanding baseline" has been derived from this application of baseline management procedures in the engineering management context.

During early stages of a CPCI development, the developer should implement those or similar procedures to control the design documentation prepared for review at PDR and CDR; later, they should be extended to include the listings. Related techniques, including the use of automated support tools and other "library controls" (see MIL-S-52779, para. 3.2.5), can be employed to control and account for elements of computer program code as those are generated and refined through successive levels of developmental testing.

Use of the label "configuration management" for techniques devised for those purposes is not infrequent; and it represents one source of potential confusion to software managers who become involved in military system programs. The confusion is not easy to dispel, since: such measures do constitute management controls; they are in fact dealing with the item's configuration; and, there are indeed some aspects of the controls which should also involve the configuration manager. From the point of view of distinctions established among Air Force acquisition management disciplines, however, the principal consideration is the fact that primary control of the process, up to the point of initial specification completion, must remain with the technical managers--consistently with their responsibility to develop an end product (the CPCI) which meets specified requirements of its Part I specification. At the same time, surveillance and support of their methods should also be furnished by others. As examples:

- The developer's quality assurance manager is responsible for assuring that controls in the areas in question are developed, internally documented, and implemented. While the specific techniques are not currently spelled out in any standards, requirements for the contractor to meet those objectives are included in MIL-S-52779 (AD).
- The configuration manager should provide and monitor the observance of internal standards in such areas as identification numbers and markings (2.4), specification requirements, and maintenance of design documents to incorporate approved changes to the Part I specification. Again, specific procedures are largely at the contractor's option.

4.5.2 Engineering Release Systems

Requirements for engineering release records to assure that proper relationships are maintained between engineering data and manufactured CIs are covered in Appendix X of MIL-STD-483. Statements are made therein (and elsewhere; cf. AFSCM/AFLCM 375-7, paragraph 1-39,j) that the specific requirements set forth for hardware do not apply to CPCIs, but that computer program contractors should implement procedures to comply with the "intent and objectives". AFR 800-14 (Vol. II, paragraph 6-6,c) suggests that the procedures apply to development as well as to production, and states that they should be "tailored to cover all CPCI documentation".

No clarifications are provided in any known source, however, to identify what the analogous procedures might actually consist of, for software. The objectives themselves are subject to varied interpretations because of their apparent orientation towards product-level controls/records associated with hardware manufacturing. Hence, in the absence of a better understanding of what kinds of actions software contractors should take to comply, it is the summary recommendation of this guidebook that program managers regard the engineering release system requirements as "not applicable" to software. Pertinent considerations include the following:

- Engineering release systems involve internal contractor controls over engineering drawings, together with records of drawing numbers, parts numbers, effectivities, etc. which relate basic requirements and engineering changes to production units of a CI. The importance of such systems stems from the significant role of the Part II specification (largely, engineering drawings) in governing the production process, and from the key importance of production in the CI acquisition cycle.
- The question of what objectives are analogous to those in software is subject to some debate, since: a computer program Part II specification does not have that role in governing CPCI "production" (tape/disc duplication); nor does the latter process have comparable significance as a portion of the overall CPCI acquisition (see 3.6).
- Study of Appendix X suggests that some of the procedures are related to document controls, tape or card deck controls, and record-keeping practices for which software requirements are recognized under labels other than "engineering release". Examples are: controls and records of changes to design documents reviewed at PDR and CDR (see 4.5.1 above); and certain functions served by document numbering practices, the configuration index, and the version description document as discussed in the next section (5.0). The latter is the area which perhaps

furnishes the most direct analogies to engineering release, since it includes records which maintain relationships, after PCA, among basic specification requirements, changes, other documents, and numbered versions of the CPCI. However, program managers will probably find it advisable to continue to handle those and similar areas on their own merits for software, disregarding whether analogies can be drawn with the hardware engineering release practices.

4.6 INTERFACE CONTROL

Interface control is primarily a system engineering/design integration, rather than a configuration management, function. Its objectives are to assure that hardware and software elements being supplied by different participating sources will fit and function effectively together when assembled into the complete system. Hence, the tasks of identifying and defining interfaces, like those of generating specifications, are basically technical. Configuration management activities associated with interface control include providing standards, procedures, and administrative support to ensure that interface agreements arrived at through technical analysis and coordination are properly reflected in baselined specifications.

Currently-available guidance and requirements pertaining to technical as well as other aspects of interface control are largely limited, however, to coverage provided in the configuration management standards. Familiarity with information contained in the sources identified below is essential to an understanding of the policies and procedures as they apply both at the general level and specifically to software:

AFSCM/AFLCM 375-7:

- 1-12 Systems Engineering/Design Integration Relationships to Configuration Management.
- 1-39,b Interface Control.

MIL-STD-483:

- Appendix II Interface Control.
- 60.4.3.1.1 Paragraph 3.1.1 Interface Requirements

4.6.1 General Concepts and Responsibilities

Interface control procedures in system programs are generally limited to interfaces at, and above, the CI level. They do not cover the control of interfaces internal to a CI, since that represents an integral part of the (single) contractor's engineering management responsibility for the CI's technical development. Further, as dealt with in the standards, the major emphasis is on interfaces involving separate contractors and/or Government agencies. Basically, the process consists of establishing and maintaining technical agreements among the different organizations responsible for interfacing systems and system elements.

In this context, an "interface" is a common boundary between two items. From the point of view of either side of the boundary, the interface implies a source of requirements and/or constraints on the configuration of the given item. Hence, when recognized and taken into account, it determines one part of the configuration defined, or to be defined, in each item's specification. An interface is "identified" when it is determined that a common boundary exists. It is "defined" when the functional and physical characteristics can be appropriately specified (or referenced) in the affected specifications.

Hence, interfaces are defined at different levels, corresponding with the levels of uniform specifications. Specifically: (a) they may be defined in functional terms at the system, segment, and CI (allocated baseline) levels, with successively increasing completeness and detail; and in addition, they may be further defined at the product level, for equipment CIs, in terms of physical dimensions, electrical or chemical etc. properties, and tolerances.

Requirements for interface control activities outlined in MIL-STD-483 apply primarily to the full-scale development phase. Interfaces analyzed and documented in the specifications prior to that time serve as technical criteria to be observed by those involved in the development phase interface control effort. Typically, the definitions existing at the end of the validation phase are incomplete with respect to matters of design approaches and responsibilities to be resolved or determined during negotiations for the full-scale development; and in addition, they require further definitions at lower levels as the design of individual CIs evolves. The latter is an important and continuing activity for equipment interfaces, in particular. Installation control--referring to equipment/facility interfaces with respect to space, locations, environment, etc.--is also a part of the interface control activities.

Interface control in a large and complex system program is accomplished by an interface control working group (ICWG) composed of members representing each contractor and Government agency involved in the program. Prime responsibility for managing, chairing, and providing administrative support to the ICWG is assigned to an interface control contractor. Other members have collateral responsibilities for defining and controlling interfaces affecting their system segments, CIs, or interfacing systems. The basic activity consists of arriving at technical interface definitions, documenting those in the form of interface control drawings (ICDs), implementing controls, and maintaining records of ICD actions.

Configuration control actions as such occur when the ICWG completes and approves individual ICDs. Affected contractors prepare coordinated ECPs and process those through the system CCB to incorporate the interface definitions into base-lined specifications. For equipment CIs, they are normally incorporated by reference, rather than directly; hence, the ICDs themselves are then used in conjunction with the specifications, together with other engineering and facility construction drawings, to control the design and subsequent integration of the CIs.

Program office planning for interface control must be accomplished during the validation phase to a level which makes it possible to clearly delineate, in development phase RFPs and statements of work, the approach to be taken and the specific responsibilities of each participant. Requirements must be tailored to the contractor's structure, complexity of the system, and complexity of interfaces with other C systems. Taking those factors into account, RFPs should identify plans for establishing the ICWG, describe its functions and composition, identify the interface control contractor, and define the scope of interfaces to be controlled at that level. Separate ICWGs below the system level are appropriate when the program involves associate contractors responsible for major system segments. Specific planning for those, as well as for participation in the system ICWG, should be included in system engineering and configuration management portions of the associates' proposals.

4.6.2 Documentation and Control of Software Interfaces

ICDs may be prepared in many forms, depending on the type of interface type of CIs involved, and the level of interface identification or definition required. For computer program interfaces (and others of a functional nature), they may take the form of "book-form" drawings. Such drawings are required to bear minimum information for identification and control purposes--such as drawing number, revision level, and date--but their format and content are not otherwise constrained. Hence, when ICDs involving computer programs are found to be necessary for ICWG uses, their content can be prepared in a form suitable for direct incorporation into the CPCI Part I specification--i.e., complying with content requirements set forth in Appendix VI of MIL-STD-483 for the interface requirements paragraph, 3.1.1.

Interface control involving computer programs should be included in the ICWG activities to the extent necessary to establish and maintain compatibility with other elements of the system as a whole. However, that involvement should be generally much more limited in scope than it typically is for equipment items and facilities, for such reasons as the following:

- All external interfaces of a CPCI with other items must be specified at the Part I specification level, or higher. This requirement stems basically from the fact that computer program external interfaces represent functional, rather than physical, characteristics--both for the given CPCI and for the interfacing other items.*
- For computer programs, interface definitions may not be incorporated into the Part I specifications by reference to ICDs. It is possible that agreements on some previously-undefined interfaces may be arrived at through ICWG efforts at an early stage of the development phase and documented in the form of ICDs. When that happens, however, FCPs/SCNs should be prepared to incorporate the contents directly into the specifications, normally by PDR, for subsequent control by the CCB. Later needs for ICWG uses of the information in the specific form of ICDs should be minimal.**

It tends to be typical that the most prominent interfaces of computer programs with other system elements, both hardware and software, are messages. And in some ways, messages represent a unique type of interface. A single message may contain elements which constitute interfaces, for a given CPCI, with both equipment (e.g., communications) and other CPCIs; and further, the interfacing software items are often remotely located in space and in time. Remoteness in

*The functional vs. physical distinction is less meaningful for computer programs than for equipment, especially when the computer programs are considered in isolation. One key to the logic in this context, however, is the fact that any equipment/computer program interface is limited to functional characteristics which have to be specified at the Part I specification level on the equipment side. For example, if the equipment processing capacities and speeds, etc. are known, such product-level properties as dimensions, construction, and materials are of no additional consequence to a CPCI developer.

**Exceptions have occurred when the Part I specifications were inadequate or missing. Under those circumstances, ICDs have been generated and used at later stages as one device to help overcome the resulting problems encountered during installation and testing.

space, for example, is typical when the messages are exchanged between interfacing C³ systems. Remote interfaces with respect to both space and time exist when recorded output data from one CPCI are later processed by another CPCI operating in a different computer. It is of some interest that, in contrast, remote interfaces are not normally recognized in conventional hardware practice as being a practical possibility--i.e., for working purposes, interfaces exist only at points of physical contact; yet that happens to be the class of interface characteristics which is often of predominant concern to activities involved in the identification and control of software interfaces.

To be adequate, detailed definitions of message interfaces must be provided at the bit/byte level, including the specification of such characteristics as format, lengths, data content and definitions, parity and/or redundancy, timing, and control. Once so defined, lower-level definitions are not needed, for purposes of guiding or constraining the CPCI developer.

As regards the practice of not specifying CPCI interfaces by reference to ICDs, it is significant that all message interfaces are also CPCI inputs and outputs, and that definitions of the latter represent essential and major portions of any CPCI's Part I specification content. The specification of interfaces, inputs, outputs, and related data base items "by reference" is permissible, internally to the specification itself. That is a device which should normally be employed in order to reduce redundancy and promote consistency of content across portions of the specification concerned with those elements. The important points to consider are that: (a) all of the information that might be also be documented on ICDs is required to be contained in the specification for other purposes; and (b) if the information does exist separately on ICDs, problems of maintaining the necessary consistency may be increased.

In addition to remote messages, other types of software interfaces to be examined and defined include: (a) with hardware, relevant functional characteristics of the computer, peripherals, and display/control consoles; and (b) functional and format characteristics of other software operating in the same computer. For a given CPCI, the existence and general nature of its interfaces with all other hardware and software items should be identified in the first interface subparagraph (3.1.1.1) of its Part I specification, preferably in the form of a schematic block diagram. Requirements for the detailed interface definitions stated in MIL-STD-483 (for subparagraph 3.1.1.2) vary as a function of each interfacing item's status as well as its nature. That is:

- In many cases, the interfacing item already exists. Examples are commercial computer, peripheral equipment, and associated support software. In these cases, the interface definition may be confined to identifying each item and referencing its existing specification.
- Detailed definitions of specific functional characteristics are required to be spelled out directly in the specification only for those interfacing items which are being developed concurrently with the given item, in whole or in part. In general, it is to this category of interfaces--i.e., where both sides of the interface are undergoing concurrent development--that most interface control activities of an ICWG and others are typically devoted.

SECTION 5. DOCUMENT MAINTENANCE AND STATUS REPORTING

This section discusses requirements and procedures for the identification and maintenance of computer program specifications and related documents, and for reporting the status of documents, change proposals, and delivered CPCIs. The procedures are directly related to, and depend on, procedures of configuration identification and control discussed in preceding sections. When properly integrated with those, they are designed to serve the following significant purposes:

- Provide devices to support and verify the systematic maintenance of specifications and other documents which depend on CPI configurations for their content.
- Maintain traceability and correlation of approved changes among all maintainable documents.
- Maintain correlation between documentation and delivered CPCIs.
- Maintain periodic reports which make the status of CPCIs and their documentation visible to controlling and participating activities.

Unlike configuration management standards in other areas, the requirements in this area are largely ones which originated specifically for software. They contain some elements which are analogous to, but which generally replace, the hardware-oriented requirements for configuration status accounting and engineering release (cf. 4.5.2). Comparisons between the hardware and software practices have proved to be frequent sources of confusion, partly because potential cross-applications of certain document control procedures or status reports are discernible on both sides. Those possibilities tend to be deceptive, however, due to timing requirements, objectives, and interrelationships with other management factors which differ fundamentally for the two classes of configuration items. Again, the differences are related to the fact that hardware procedures are based primarily on conditions associated with production and logistic support, whereas the software practices in this area emphasize development (or redevelopment) as being the process of major management concern during a CPI life cycle.

Guidance and formal requirements pertaining to topics addressed in this section are to be found in identified parts of the source documents listed below:

AFSCM/AFLCM 375-7:

- 1-39,h Specification Maintenance
- 1-39,i Document and Item Identification Numbering

MIL-STD-483:

- 3.4.9 Specification Authentication
- Appendix VIII Specification and Support Document Maintenance,
Computer Program
- Appendix IX Document and Item Identification Numbering and Marking

MIL-STD-490:

- 3.2 Style, Format, and Identification of Specifications
- 3.3 Changes and Revisions

Among the above, the major source of requirements specific to software is Appendix VIII of MIL-STD-483, which covers computer program SCNs, status reporting, and the version description document. Other references contain "bits and pieces" of standard requirements for document identification and maintenance which normally apply to software as well as to hardware specifications. While the standards are basically sound, they have often proved difficult to use because of their scattered locations, inadequate explanations, and some inconsistencies. However, they have also proved to be indispensable to effective software acquisition management when properly understood and used. Specific problem areas are identified, where they exist, in the subsections below; otherwise, the content of this section is consistent with the standards as they are currently written.

5.1 DOCUMENT IDENTIFICATION AND MAINTENANCE

This topic encompasses requirements for numbers and related identification practices which apply to basic issues, change page issues, and revisions of computer program specifications and other maintainable documents that are significant to configuration management activities. As is true of other areas, close coordination is required with the data management function. In this case, the requirements are imposed and monitored by configuration management, but must be implemented by data management activities and included in (and occasionally reconciled with) the developer's internal standards/procedures for that function.

Specific requirements for document identification and maintenance contained in the military standards are limited to specifications. While those are usable, they are generally insufficient to meet needs encountered in software configuration management:

- The standards referred to are those which cover minimum requirements designed for basic hardware specifications, but excluding their associated engineering drawings. The additional coverage which is available for the latter, in some abundance, is not applicable to computer programs. Yet similar needs exist for control and traceability of CPCI characteristics which are documented wholly within the content of the specifications themselves.
- Computer program specifications tend to be voluminous, partly because they do not depend on references to engineering drawings and for other reasons. As for equipment specifications, the maintenance procedures should be designed to assure that changes are accurate, complete, and traceable. But frequent changes affecting many parts can also place unique demands on their efficiency, with respect to such factors as speed and economy of handling.
- Responsibilities of software configuration managers include status keeping and reporting for all deliverable and maintainable documents which may be affected by approved ECPs, as well as for specifications, ECPs, and CPCIs. Coverage of identification and maintenance practices which apply to those other documents, in addition to the specifications, is needed to support that purpose.

Thus, in the subparagraphs below, requirements defined in the military standards are referenced, but additional requirements are identified which the standards do not currently define for uniform application. A software developer's internal standards should provide for those in some suitable manner, since adequate provisions for efficient document identification and maintenance are essential to meeting the needs of software management in other areas. Topics to be covered which are of interest to configuration managers are summarized briefly as follows:

- Definitions and procedures pertaining to types and forms of document issues, including: drafts; basic issues; change page issues; revisions; and document series (multiple volumes).
- Document numbering systems.
- Rules for identifying pages and change pages.
- Standard formats and identification rules for special front-matter pages (i.e., title page, specification change notice, document change notice, list of effective pages); and rules pertaining to the use of those pages as they apply to basic issues, change pages issues, revisions, and volumes.

NOTE: Special rules specified in DoD 5220.22-M must be observed in marking and handling documents which are classified. However, those are not addressed in this guidebook. A few of the procedures described herein for efficient maintenance and accountability of less sensitive documents do not apply to accountable SECRET documents, e.g., with respect to reissuance of title pages.

5.1.1 Document Issues

The documents that are of interest for purposes of this discussion are those which are subject to being reissued in some form when affected by approved ECPs. They consist of the specifications, test plans, and other documents to be listed for status reporting in the computer program configuration index (see 5.2). Rules for identification and handling should provide for distinguishing the various forms in which issues and revisions may occur as listed below.

- Single-Volume vs. Document Series. Any document identified by a single document number which is issued in the form of multiple volumes, including separately-bound appendices, is a document series. The document series applies whether the separate volumes are issued concurrently, as for a specification, or sequentially; examples of the latter are the status reports or version description document, for which successive issues are often identified (in each case, separately) as successive volumes of a single series.
- Draft vs. Basic Issue. A given document may undergo some number of issues, reissues, and/or corrections in draft form. Its basic issue is the initial issue prepared for formal delivery in approved form, normally following a review cycle based on the draft.*
- Revisions vs. Change Page Issues. A revision is a complete reissue of an entire document which supersedes all pages of any preceding issue.** Modifications to computer program documents, particularly specifications, are normally accomplished by issues of change pages, except when complete revisions are specifically directed by the procuring activity. Formal modifications in other forms, e.g., errata sheets, should not normally be permitted, particularly for specifications.

*CDRLs regularly designate that issue as the "final". However, "basic issue" is a more realistic label for the role it actually acquires, in the usual case.

**That definition is established for specifications in MIL-STD-490 (paragraph 3.3.1). "Revision" is also used in MIL-STD-490 and elsewhere as the general term to cover any kind of a document change.

5.1.2 Document and Page Identification

Requirements set forth jointly in MIL-STD-483 and MIL-STD-490 for identifying numbers and other information to be provided on the title page and each other page of a specification are summarized in Table 5-1. A few other numbers are also shown which are not required in those standards, but which have been found useful in software contracts to identify a particular volume, appendix, or specification part and to provide a positive link with SCNs (or other change issue identifier).

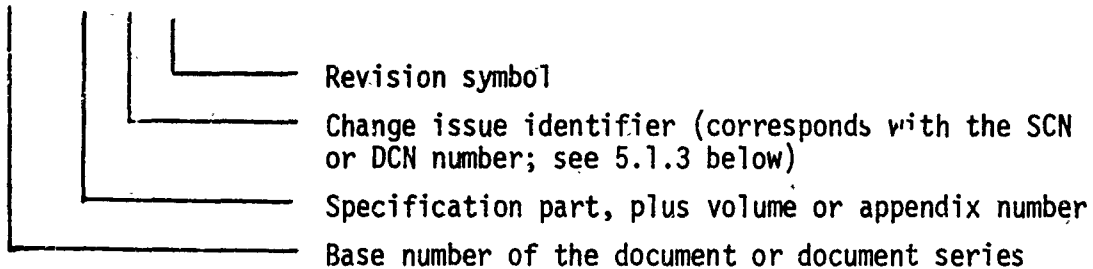
The numbering of documents to include various useful elements of information can obviously be accomplished in many ways. One example is shown below, based on a numbering system which was developed and has been used specifically for handling software documents. It provides all of the needed elements in a

Table 5-1. Summary of Identification Data Required for CPCI Specifications.

<u>TITLE PAGE</u>	(1)	<u>OTHER PAGES</u>	(6)
SPECIFICATION NUMBER	(2)	SPECIFICATION NUMBER	(6)
REVISION SYMBOL	(3)	REVISION SYMBOL	(6)
CHANGE ISSUE IDENTIFIER	*	CHANGE ISSUE IDENTIFIER	*
VOLUME NUMBER	*	VOLUME NUMBER	*
SPECIFICATION PART	(4)	SPECIFICATION PART	*
DATE	(1)	DATE	(6)
CODE IDENTIFICATION	(2)	MARKINGS	(7)
SPECIFICATION TITLE	(5)		
CPCI NOMENCLATURE	(5)		
CPCI NUMBER	(5)		
AUTHENTICATION	-(1)		
(1) MIL-STD-483, paragraph 3.4.9 and Figure 1 (2) MIL-STD-490, " 3.2.16.2 (3) MIL-STD-490, " 3.2.16.3 (4) MIL-STD-490, " 3.2.16.4 (5) MIL-STD-490, " 3.2.16.7 and Figure 1 (6) MIL-STD-490, " 3.2.16, 3.3.2, 3.3.3 (7) MIL-STD-490, " 3.3.2.2, 3.3.2.4, 3.3.3 * Elements not specified in the current standards			

relatively simple and efficient form, although it does not comply literally with certain format details specified in the standards for specifications:

9999-999-99 X (Complete document number)*



Requirements for numbering volumes and appendices on title pages, and for arrangement of the volume/appendix title, are not clarified in the standards directly for titles of CI specifications. Titling is generally accomplished in the same manner as described in Appendix III of MIL-STD-483 for system segment specifications. Volumes of a specification are numbered in Arabic numerals, beginning with "1". Appendices are numbered in Roman numerals, beginning with "I". Example:

COMPUTER PROGRAM DEVELOPMENT SPECIFICATION
for
ORBIT PREDICTION
CPCI No. 3021900
Volume 5. ELEMENT COMPARISON
[or: Appendix II. CLASSIFIED SUPPLEMENT]

5.1.3 Front-Matter Pages

A title page is normally the first page of front matter in the basic issue of any document or volume, whether or not a hard cover is also provided. Since it bears the full document identification, including the issue date, a new title page should be issued as a part of each change page package.

In addition, each change page issue to a specification must be accompanied by a specification change notice (SCN). The SCN functions, in part, as the change page cover which accompanies the FCP for review and approval by the procuring

*"9" = numeric; "X" = alphabetic

activity CCB prior to being distributed. It also functions, however, as a special page of front matter to be inserted into each copy of the specification, since a copy of the approved SCN is included in the set of change pages distributed to each holder of the specification. The sample format and basic instructions for preparing SCNs outlined in MIL-STD-490 are supplemented for computer programs in Appendix VIII of MIL-STD-483. Among various additional clarifications which have been found useful are the following:

- Successive SCNs against a given specification are numbered in sequence, beginning with "1" for the first SCN issued against the computer program development (Part I) specification. A separate sequence of SCN numbers, again beginning with "1", applies to SCNs for the computer program product (Part II) specification.
- When the specification is issued in the form of separately-bound volumes or appendices, one SCN form is prepared for the change page issue to each affected volume or appendix, and is identified by a dash number consisting of (a) the appropriate sequence number of SCNs for that specification, followed by a dash and (b) the number of the given volume or appendix. (Examples: 23-2, or 23-IV).
- It is essential that each SCN issued to incorporate a Class I change also incorporate all Class II changes accomplished since the preceding issue of the specification or modification thereto. Class II changes are identified individually on the SCN, in addition to being reported on Class II CRs submitted with the given ECP/SCN package.
- If a complete revision incorporates one or more ECPs not previously implemented through SCNs/change pages to the preceding issue, an SCN should be included as an integral part of the revised issue to identify those ECPs as being incorporated.

In practice, some program managers have permitted certain latitude in the format and preparation of computer program SCNs to facilitate the processing of high-volume changes. One useful device is to substitute a list of effective pages (LEP) for the "Summary of Previously-Changed Pages" portion of a standard SCN.* Since that device appears to be in process of becoming a formally-recognized option for computer programs, as indicated in a current coordination draft of MIL-STD-490A, its use is discussed further below.

*Such otherwise-trivial considerations can become relatively significant in programs like the one described in ESD-TR-69-302 (Searle et al.; see 8.2), in which change pages were issued to one Part I CPCI specification at an average rate of 200 per month over a 29-month period, incorporating an average of more than 2 Class I and 4 Class II changes per month.

Information provided by the LEP is indicated in Figure 5-1 (top). The basic issue of a document contains a listing of page numbers only, in the first column. With each issue of change pages, entries are added in the second and third columns to show the SCN or DCN (see below) number and issue date of the package. As succeeding issues occur, entries shown on the last preceding issue are retained for all pages that remain unchanged by a new issue. Thus, the LEP contains a complete account of the current status of the given volume. Accordingly, when it is used in that manner, the printed statement on accompanying SCNs should be changed from that illustrated in Figure 3 of MIL-STD-490 to read as follows:

"This notice informs recipients that the specification identified by the number shown in the 'SPEC. NO.' block above has been changed. The pages changed by this SCN are those furnished herewith and carrying the same date as shown in Block 12 above of this SCN. The pages of the numbers and dates listed in the accompanying list of effective pages constitute the current version of this specification."

The document change notice (DCN) serves essentially the same functions for other maintainable documents that the SCN serves for specifications, in that it provides a record of status relative to incorporated ECPs which is contained directly in each copy of the document. A sample format is illustrated in Figure 5-1. The DCN is useful for the test plan, test procedures, handbooks, and manuals listed in the configuration index. It does not apply to the version description document, since each issue of a VDD is a new document which includes listings of incorporated changes in its content. Uses of DCNs are similar to those of SCNs. However, it should be noted that:

- Requirements for such a form are not explicit in the standards. Its use is suggested in this guidebook as one device to support data and configuration management requirements implied by the configuration index.*
- Class II changes do not apply to non-specifications; and, changes may occur to those documents both as a result of ECPs and for reasons unrelated to configuration management. That is: test plans, handbooks, and manuals are subject to change for technical and other reasons, in addition to impact by ECPs. Configuration managers track and report all updates to those documents because some of them do result from ECPs and therefore provide indicators of ECP implementation. But the ECPs and CRs are processed directly only against the specifications.

*It has been noted that "DCN", if adopted officially, might conflict with the "design change notice" used in configuration management of equipment items. This guidebook is recommending only that a developer should provide that kind of information, not that the form necessarily carry that label or be prepared in any standard format. "DCN" was chosen here only for convenience of discussion, and because of its obvious similarity to "specification change notice".

(Issue Date)	(Full Document No.)				
LIST OF EFFECTIVE PAGES					
Date of Basic Issue:					
PAGE NO.	CHANGE IDENTIFIER	ISSUE DATE	PAGE NO.	CHANGE IDENTIFIER	ISSUE DATE

Computer Program
SPECIFICATION CHANGE NOTICE

1. ORIGINATOR NAME AND ADDRESS		DATE PREPARED	
		2. ☐ PROPOSED ☐ APPROVED	3. CODE IDENT. 4. SPEC. NO. 5. CODE IDENT. 6. SCN NO.
7. SYSTEM DESIGNATION	8. RELATED CP NO.	9. CONTRACT NO.	10. CONTRACTUAL AUTHORITY
11. CONFIGURATION ITEM NOMENCLATURE		12. EFFECTIVITY	

THIS NOTICE INFORMS RECIPIENTS THAT THE SPECIFICATION IDENTIFIED BY THE NUMBER SHOWN IN THE "SPEC. NO." BLOCK ABOVE HAS BEEN CHANGED. THE PAGES CHANGED BY THIS SCN ARE THOSE FURNISHED HERewith AND CARRYING THE SAME DATE AS SHOWN IN BLOCK 12 ABOVE OF THIS SCN. THE PAGES OF THE NUMBERS AND DATES LISTED IN THE ACCOMPANYING LIST OF EFFECTIVE PAGES CONTAIN THE CHANGES.

13. FCP/C[illegible]

Figure 5-1. Sample Front-Matter Pages for Maintainable Documents.

5.2 CONFIGURATION INDEX

The computer program configuration index (or simply, "the index") is one of two periodic software status reports required for configuration management, the other being the change status report. The two should be issued concurrently. Together, they present information which permits users and managers to monitor the status of documents, events, and changes. They also lend themselves to cross-checking for consistency with each other, as well as for consistency with such other sources as ECPs and version description documents. A major distinction to be kept in mind is that the change status report is concerned with ECPs, directly, whereas the index reports the status of individual documents.

Both the index and change status report have been "automated" in some past system programs, in that they have been issued as computer printouts of status data stored on tape. However, manual preparation may often be more cost-effective, particularly during early stages of a program. Neither report involves computations or other complex data manipulation. Both do involve:

- Establishing orderly files of status data, organized into identifiable records.
- Updating the files selectively--i.e., adding, replacing, or deleting data.
- Provisions for audit--i.e., verifying the data updates with respect to such factors as timing, source, and accuracy.
- Selective retrieval and printing of the data in required reporting formats.

The purpose of the index is to provide a record of specifications and other maintainable documents issued to support the development and use of a computer program configuration item. Its principal direct functions are to (a) report the basic issue or any complete revision of each maintainable document and (b) regularly report the current status of each with respect to subsequent modifications resulting from approved ECPs. To support those functions, it also identifies approved ECPs which will affect each document, but for which modifications to the document have not yet been issued. Additionally, it contains a one-page, summary record of the dates on which developmental milestones for the CPCI are scheduled and accomplished.

Information provided in the index has proved to have important uses for the responsible developers as well as to the program office and participating agencies. Its full significance is often not apparent during early stages of

a program, since its content does not begin to expand appreciably until after the documents reported on have been formally issued. However, experience has been that users become increasingly dependent on its status information as the program progresses. Perhaps equally important is the fact that a developer's ability to issue the index, adequately, presupposes that he has effective working procedures for generating the subject documents, processing and reporting change proposals, maintaining the documents to reflect the approved changes, and maintaining accurate records of document, CPCI, and change status--all as integrated parts of his software management effort.

Unfortunately, proper implementation of the index has been handicapped in recent years by the fact that the MIL-STD-483 instructions are subject to certain conflicting interpretations. The problem, summarized very briefly, is that they appear to require the Part 1 of each major section to perform functions which are not readily compatible with some of the stated objectives for its organization and content.* It is hoped that those discrepancies will be resolved in a forthcoming revision of the standard, along lines suggested by the treatment herein. As interim measures, it is recommended that POs consider:

- Using GDRL backup instructions similar to those illustrated in Figures 5-4 and 5-5 below, to clarify the DID (DI-E-3122).
- Making associated changes to the DID for the change status report (DI-E-3123), to clarify its coverage and add other requirements outlined in 5.3 below.

Thus, the description of the index provided herein assumes those modifications to the instructions for paragraph 80.10.4.1 of MIL-STD-483 and its associated Figure-13. In other respects, it is consistent with the instructions as written.

5.2.1 Organization and Timing

A specific format for the configuration index is not mandatory, and formats can be expected to vary. The required general organization includes a title page, table of contents, and the following series of sections:

*For a further discussion of the questions which have been raised about this area, see 7.1

- Section A - CPCI Development Record
- Section I - CPCI Development Specification
- Section II - CPCI Product Specification
- Section III - CPCI Test Plan/Procedures/Reports
- Section IV - Handbooks
- Section V - Manuals
- Section VI - Version Description Document

When the given developer is responsible for a group of related CPCIs, consideration should be given to the option of preparing one index for the group as a whole. That option has certain advantages when some manuals or handbooks tend to be related to the group rather than one-to-one with individual CPCIs. In that case, a suitable arrangement is to group all other sections by individual CPCI, in order, and to provide a common Section IV and/or V at the end.

The requirement is to initiate the index within 30 days following the date of basic issue of the Part I specification for the CPCI being reported. It is issued periodically (as specified in the CDRL, normally each month) thereafter. The initial issue for a single CPCI will typically consist of only four pages --namely, the title page, table of contents, Section A, and Section I. The report expands in size, as a joint function of ECPs and the addition of other sections, as the development phase proceeds.

5.2.2 Preparation of Sections

Samples illustrating the forms in which information can be provided in some of the sections are shown in Figures 5-2 through 5-7. Except for instructions pertaining to the Part I of Sections I through VI, as mentioned above, the minimum preparation requirements as described in Appendix VIII of MIL-STD-483 are generally self-explanatory. Again, however, internal policies and procedures should be carefully formulated and documented by the developer. Some questions can be encountered which may have to be resolved by the program office, particularly if the program involves multiple software developers. As examples:

- The listing of each document or volume generally begins with its basic issue, but some exceptions may be indicated. For example, manuals and handbooks are often issued and formally delivered for use during system DT&F in "draft" or preliminary form. If ECPs affecting those are likely to be implemented in the interim, the record of those draft issues and their modifications should be reported in the index.

SECTION A. CPCI DEVELOPMENT RECORD

CPCI NO. AND NOMENCLATURE		CPCI No. 3021000 SSC Executive and Man-Machine Interface Computer Program for Space Surveillance Center, System 499L	
DEVELOPMENT SPECIFICATION NO.	ESD499L3021000		
ISSUE DATE	30 Jan 76		
AUTHENTICATION DATE	30 Feb 76		
PRELIMINARY DESIGN REVIEW SCHEDULE		CRITICAL DESIGN REVIEW SCHEDULE	
15 Apr 76 15 Apr 76 C		11 Feb 77 12 Feb 77 C	
FUNCTIONAL CONFIGURATION AUDIT SCHEDULE		PRODUCT SPECIFICATION SCHEDULE SUBMITTAL	
17-Sep-77 24 Sep 77 24 Sep 77 C		25 Nov 77	
PHYSICAL CONFIGURATION AUDIT SCHEDULE		CPCI QUALIFICATION SCHEDULE	
8 Mar 78		14 Oct 77	
PRODUCT SPECIFICATION SCHEDULE AUTHENTICATION		CPCI QUALIFICATION CERTIFICATION DATE	
1 Apr 78		8 Mar 78	
QUALIFICATION TEST DOCUMENTS		See Section III of this Configuration Index	
CONTRACTOR		Electronic System Developers, Inc.	CONTRACT NO E19628-78-C-0000

Figure 5-3. Configuration Index: Sample Section A

COMPUTER PROGRAM CONFIGURATION INDEX		Page
SECTION A.	CPCI DEVELOPMENT RECORD	1
SECTION I.	CPCI DEVELOPMENT SPECIFICATION, ESD30206	
	Part 1. Basic Documentation	2
	Vol. 1. General	2
	Vol. 2. Startup & Executive Control	2
	App. 1. Classified Supplement	3
	Part 2. Approved Changes	3
SECTION II.	CPCI PRODUCT SPECIFICATION, ESD30206	
	Part 1. Basic Documentation	4
	Vol. 1. General Design	4
	Vol. 2. Information Transfer	4
	Vol. 3. Displays	4
	App. 1. Listings	4
	Part 2. Approved Changes	4
SECTION III.	CPCI TEST DOCUMENTS	
	Part 1. Basic Documentation	5
	Test Plan, TR 1600-001	5
	FQT Procedures, TR-1600-101	5
	FQT Report, TR-1600-201	5
	Part 2. Approved Changes	6
SECTION IV.	HANDBOOKS	
	(Not applicable to this CPCI)	
SECTION V.	HANDBOOKS	
	(Basic Issue scheduled 11-12-78)	
SECTION VI.	VERSION DESCRIPTION DOCUMENT, TR-1084-xxx	
	Part 1. Basic Documentation	7
	VDD-1, TR-1034-001	7
	VDD-1A, TR-1084-002	7
	Part 2. Approved Changes	7

Figure 5-2. Configuration Index: Sample Table of Contents.

DI-E-3122

Make the indicated substitutions for Figure 13 and paragraph 80.10.4.1 of MIL-STD-483:

Fig 13. Reference the sample format attached herewith in preparing the Part 1, Basic Documentation, of Sections I through VI.*

Para 80.10.4.1. The Part 1 of each section shall contain information equivalent to that illustrated in the attached figure. The figure illustrates data only for Section I of the index, and for a development specification which is issued as a series of separate volumes and appendices. Data shall be provided in the Part 1 of each section as appropriate to single-volume documents and the given section, as follows:

a. Issue. The first entry in this column is always "BASIC" for all documents or volumes in Sections I through V. The first entry in Section VI is "VDD-1". Each succeeding entry will be:

(1) for Sections I and II, an SCN number; or

(2) for Sections III through V, a change issue identifier; or

(3) for Section VI, a version description document (VDD) number.

Except for VDDs in Section VI: "BASIC" is replaced by a suitable indicator when a complete revision is issued, e.g., "REV. A". The listing of all previous updates to the basic issue or to any previous revision is deleted from the index when a new revision appears.

b. Change. For the first entry, this column shall be used to identify the number of the given document or volume. For each succeeding entry in the "ISSUE" column (i.e., SCN number, change issue identifier, or VDD number), this column shall contain a listing of the numbers of all ECPs which are incorporated in the given updating. In addition to Class I ECPs, Sections I, II, and VI shall identify the numbers of all Class II changes incorporated.

c. Title. The title of the given document or volume is entered opposite the "BASIC" entry. The title of the change is listed opposite each ECP or Class II change report number appearing in the preceding column.

d. Date of Issue. The date listed is the date of issue corresponding to each basic issue, SCN, change issue identifier, or VDD number appearing in the first column.

*[Ed. Note: Attach a figure similar to the adjacent figure —>]

Figure 5-4. Configuration Index: Sample
CDRL Backup Instructions.

COMPUTER PROGRAM CONFIGURATION INDEX				
SECTION I - COMPUTER PROGRAM DEVELOPMENT SPECIFICATION, ESD499L385 (9 VOLUMES + 4 APPENDICES)				
PART 1. BASIC DOCUMENTATION				
ISSUE	CHANGE	TITLE	ISSUED	
BASIC	VOLUME 1	GENERAL	01-31-76	
SCN 1-1	ECP 1	F/TELL OVER MULTIPOINT LINE	02-30-76	
	CR 1	CLARIFY FUNCTION DESCRIPTION		
SCN 4-1	ECP 7	DELETE BIRDIE-6	05-07-76	
	:	:	:	
BASIC	VOLUME 2	SURVEILLANCE	01-31-76	
SCN 1-2	ECP 1	F/TELL OVER MULTIPOINT LINE	02-30-76	
SCN 3-2	ECP 4	ADD TRACKING-ON-STATION ARPS	04-06-76	
	CR 3	DELETE REFS IN TABLE 2-1		
	CR 5	CLARIFY ALRI SMOOTHING CONSTANTS		
SCN 9-2	ECP 11	SYSTEM LIMITS FOR MANUAL REPORTS	12-05-76	
	:	:	:	
APPENDIX IV (continued)				
SCN 38-IV	ECP 57	MODIFY SUPPRESS TAPE SRN MESSAGE	03-18-77	
	CR 50	CLARIFY SIM TAGS		
SCN 47-IV	ECP 63	CHANGE EVALUATION REQUEST FORMAT	06-12-78	
	CR 60	MISCELLANEOUS CORRECTIONS TO SIM		
PART 2. APPROVED CHANGES				
ECP NO.	TITLE		VOL/APP	APPROVED
90	EXPAND LEGAL RANGE OF TAPE SRN		1, 5, 7, I, III	12-19-77
94-1	MONITOR EXCHANGE TELL LINE		3, 4	02-06-78
95	COMPENSATE GRID VS. TRUE NORTH		1, 2, 3, II	06-24-78
101	DISPLAY FUNCTION CHANGES		3, 8, 9, I	06-24-78
105	DELETE AUTOMATIC MODE CHANGE		2	07-02-78

Figure 5-5. Configuration Index: Sample
Section I.

COMPUTER PROGRAM CONFIGURATION INDEX				
SECTION III - CPCI TEST PLAN/PROCEDURES/REPORTS				
PART 1. BASIC DOCUMENTATION				
ISSUE-	DOC/ECP NO.-	TITLE		-ISSUED-
BASIC	TR-1409-010-XX	TEST PLAN, CPCI NO. 2021031		10-16-73
DCH 01	ECP 2	CHANGE SITE LIMITS		12-10-73
	ECP 3-2	ADD DSP INPUTS		
	ECP 4-1	COMBINE COMMIT ACTIONS		
DCH 02	ECP 7	ADD SIM-2 FUNCTIONS		01-11-74
:	:	:		:
BASIC	TR-1509-101-00	TEST PROCEDURES, PQT #1		06-11-74
BASIC	TR-1509-102-00	TEST PROCEDURES, PQT #2		06-28-74
BASIC	TR-1509-103-XX	TEST PROCEDURES, PQT		08-07-74
DCH 01	ECP 43	CHANGE DEFERRED COMBAT STATUS		09-07-74
:	ECP 48-9	REVISE LAUNCH WARNING PRIORITIES		:
:	:	:		:
BASIC	TR-1509-201-00	TEST REPORT, PQT #1		07-15-74
BASIC	TR-1509-202-00	TEST REPORT, PQT #2		07-31-74
:	:	:		:
PART 2. APPROVED CHANGES				
ECP NO.	TITLE	TR AFFECTED		APPROVED
72	AUTOMATED WEATHER INPUTS	1409-010, 1509-103		08-19-74
96-1	REVISED SIF CODE	1409-010, 1509-103		09-20-74
104-3	CHANGE READINESS LEVELS	1409-010, 1509-103		09-27-74

Figure 5-6. Configuration Index:
Sample Section III

COMPUTER PROGRAM CONFIGURATION INDEX				
SECTION VI - VERSION DESCRIPTION DOCUMENTS				
PART 1. BASIC DOCUMENTATION				
ISSUE-	DOC/ECP NO.-	TITLE		-ISSUED-
VDD-1	TR-1084-001	VERSION DESCRIPTION DOCUMENT		09-26-67
VDD-2	TR-1084-002	VERSION DESCRIPTION DOCUMENT		01-16-68
	ECP 43	ADD PASSIVE TRACKING CAPABILITY		
	ECP 47	RESET ARMAMENT CONSTANTS		
	ECP 50-6	CHANGE SIM FUEL MONITORING		
	ECP 59-4	SIM/LIVE TARGET STATUS - BOMBING		
	ECP 61	SIM JAM INDICATOR FOR HEIGHT		
	CR 73	CORRECT STROBE MISC ERROR		
	CR 74	CORRECT ACCELERATION TO CRUISE ERROR		
	CR 83	MISCELLANEOUS ERRORS - FINAL PQT		
VDD-2A	TR-1084-003	INTERIM VERSION DESCRIPTION DOCUMENT		01-31-68
:	ECP 79-1	SIM/CDS-OP 360 DEGREES		
:	CR 83	PASSIVE ITEM HOUSEKEEPING		
:	CR 84	TRACK CROSSING ANGLE ERROR		
:	CR 85	STARTOVER LOGIC FOR SIGOL		
:	:	:		:
VDD-4	TR-1084-007	VERSION DESCRIPTION DOCUMENT		07-15-68
	ECP 91-1	INTERCEPTOR ADA/NAV STATUS		
	ECP 102	EXPAND TAPE SEARCH RANGE		
	CR 143	ARMAMENT ROUTINE CORRECTION		
	CR 144	COORDINATE CONVERSION SIGNS		
PART 2. APPROVED CHANGES				
ECP NO.	TITLE		VERSION	APPROVED
97-5	ADD AWACS INTERFACE CAPABILITY		5	02-12-68
116-2	DESIGNATE MASTER MCC IN MULTI-MCC DIVISION		4A	05-10-68
118	DELETE AUTOMATIC CHANGE FROM ACTIVE TO MONITOR		4B	06-09-68
122	MODIFY CAT SITE SELECTION LOGIC		5	06-09-68

Figure 5-7. Configuration Index:
Sample Section VI.

- The listing of prior SCNs/ECPs to a specification is normally deleted when a complete revision is issued. If the revision incorporates any ECPs for which SCNs were not previously issued, however, the SCN contained in the revision itself (see 5.1 3 above) should be shown, beginning with the first issue of the index in which the revision is listed to replace the basic issue (or earlier revision if that should ever occur). Similar rules can apply to non-specifications.
- Although alterations in content of the index from month to month tend to be only partial, each successive issue should be a complete reissue rather than a change-page update. The exception is that if no change occurs during a given report period, a one-page negative report can be substituted.

5.3 CHANGE STATUS REPORT

The change status report is a periodic report which lists, and summarizes current status for, ECPs pertaining to CPCIs for which a given prime or associate contractor is responsible. It is supplementary to the configuration index. It is initiated following initiation of the given contractor's first ECP to the allocated baseline and is published concurrently with the index thereafter, usually at monthly intervals.

The direct function of the change status report is to disseminate information relating to the status of all ECPs which are active at a given time--i.e., which are in varying stages of preparation, processing, or implementation. Each ECP is entered in the report following assignment of a number to its preparation, and continues to appear in the report for at least one issue following either (a) disapproval by the procuring activity CCB or (b) completion of its implementation.

Instructions provided for the change status report in MIL-STD-483 are relatively clear as regards minimum requirements.* The description below incorporates all of those minimum requirements, but expands on them in two ways: (a) Whereas the basic description of the report in MIL-STD-483 is presented for "a CPI", the description below outlines the common practice of requiring one report per contractor. (b) It includes certain related and additional features which would also have to be specified via CDRL backup instructions if desired for a given program; those are:

*The title of paragraph 80.11 should be "Change status report" instead of "Change status listing". However, that error can usually be detected because of its conflict with the text and the title of the DID (DI-E-3123).

- A title page.
- Identification of CPCI numbers in Section I.
- Identification of related ECPs across contractors in Section I.
- Inclusion of the CPCI number and a summary document update schedule as part of the summary status data reported for each ECP in Section II.

Although not specified, a title page should be provided containing information equivalent to that required for the configuration index (paragraph 80.10.2 of MIL-STD-483). When prepared on a one-per-contractor basis rather than for a single CPCI, the title page should identify all CPCIs for which ECP status information is being reported. Additionally, the report consists of two major sections:

Section I. Change Status Listing

Section II. Change Status Summary

5.3.1 Section I. Section I consists of a listing of all current ECPs by ECP number, CPCI number, ECP title, status indicator, and comment (optional). If the list becomes extensive it may require more than one page. The legend for status indicators must appear in a convenient location on the first page. Referring to the sample data shown in Figure 5-8, points to be considered in the contractor's rules for preparing this section include the following:

- Each ECP number may consist of its base sequence number, a dash number (for related ECPs), a revision element, and a correction element. Numbers listed in Section I should contain at least the first three of those elements, where they have a value. ECPs are listed in Section I in order of their base number, and in order of dash number within a given base number.
- Considering various contingencies, specific rules are needed for the use of status indicators. For example:

P is the entry for an ECP at the time it is first listed unless preparation and submittal have both occurred during the reporting period. If coordination with other contractors is required following actual preparation but prior to submittal, the "P" is retained until submittal has occurred.

S applies only if the report is issued after submittal but prior to notification of the procuring activity's CCB action.

CHANGE STATUS REPORT (Computer Program)				
SECTION I. CHANGE STATUS LISTING				
ECP NO.	CPCI NO.	ECP TITLE	STATUS	COMMENT
11-R1	SD49924	DISPLAY FORMATS COMPLETED	A	
12	SD49924	DSP INTERFACE DEFINED	A	
15	SD49927	RECORD TRACK INTERRUPTS	A	
16	SD49926	SURPRESS TAPE SEN MESSAGE	X	STUDY PENDING
18-R1	SD49924	PROCESS SIM A-LINK DATA	A	BASIC ECP
18-1	SD49926	PROCESS SIM A-LINK DATA	A	
18-2	SD49927	PROCESS SIM A-LINK DATA	A	
BAC 13	BAC2750	SIM A-LINK INTERFACE	A	IMPACT BY 18-R1*
19-R2	SD49924	ACKNOWLEDGE AHACS Z/TELL	S	IMPACT BY RAC 16*
20	SD49925	DATA LINK BUFFER OUTPUT	D	
21	SD49924	DATA MONITOR CONTROL	P	
22	SD49924	POSITIVE ACTION FEEDBACK	P	
STATUS INDICATORS: P = ECP IS BEING PREPARED S = ECP HAS BEEN SUBMITTED A = ECP HAS BEEN APPROVED D = ECP HAS BEEN DISAPPROVED X = ECP HAS BEEN DEFERRED I = ECP HAS BEEN IMPLEMENTED				

Figure 5-8. Change Status Report: Sample Section I

CHANGE STATUS REPORT (Computer Program)				
SECTION II. CHANGE STATUS SUMMARY				
a.	<u>ECP NO./TITLE</u>	63-R2-CI	Process Simulated CORBA FLAME Inputs	
b.	<u>CPCI NO./TITLE</u>	3021101	Input/Output Processing (IM-355)	
c.	<u>SUMMARY OF PROBLEM:</u>	[Short narrative paragraph summarizing Block 17 of the ECP]		
d.	<u>DESCRIPTION OF PROPOSED SOLUTION:</u>	[Short narrative paragraph summarizing Block 16 of the ECP]		
e.	<u>REFERENCE DOCUMENTS:</u>	[Listing of references by numbers, titles, and dates]		
f.	<u>PREPARATION STATUS:</u>	[Short sentence summarizing the status of preparation or revision]		
g.	<u>ACTION STATUS:</u>	[Short sentence summarizing latest action accomplished by PO CCM]		
h.	<u>IMPLEMENTATION STATUS:</u>	[Applies only to approved ECPs: Statement of status, including problems, plus status of documents identified in Blocks 8 and 31 of the ECP; see sample below]		
	<u>DOCUMENT</u>	<u>PROPOSED</u>	<u>ISSUED</u>	
	ESD499L3021101	PART I SPECIFICATION	04-12-75	01-14-75
	TR-1919-001-02	CPCI TEST PLAN	05-27-75	05-27-75
	TR-1919-101-00	CPCI TEST PROCEDURES	06-08-75	
	ESD499L3021101	PART II SPECIFICATION	07-29-75	

Figure 5-9. Change Status Report: Summary of Section II Content.

A applies for all issues following notification of CCB approval, up to (not including) the first issue after implementation of the ECP is complete.

X applies for all issues following notification of a deferral action by the CCB, until another action has been taken.

I applies for one issue only, after implementation is complete. Thereafter, the ECP is deleted from the listing.

- Section I should identify each ECP which either (a) impacts another contractor's configuration item(s) or (2) is caused by impact of another contractor's ECP. (Examples are indicated by the asterisked comments in Figure 5-8.) Reporting of other-contractor ECPs is feasible when confined to Section I, at the level indicated. Further data pertaining to individual ECPs--e.g., with respect to the status of impacted documents--should be available in the Section II of the status report issued by each responsible contractor.

5.3.2 Section II. The second section contains a brief status summary in narrative or other suitable form for each ECP listed in Section I. Figure 5-9 illustrates the elements of information, which should normally not require more than one page per ECP. Considerations pertaining to two of the elements, which were mentioned above as requiring backup CDRL instructions if needed in a given program, are as follows:

- Identification of the CPCI is not called for in MIL-STD-483, but is pertinent when the contractor preparing the report is responsible for more than one developmental CPCI.
- The listing of scheduled vs. actual updates of impact documents provides a direct indicator of implementation status for approved ECPs, since computer program changes are fully implemented, in effect, when those updates are complete. The need for a listing of that general nature is recognized in MIL-STD-483, but the requirement was mistakenly imposed on the configuration index (Figure 13). Moving the requirement to this section of the change status report, in conjunction with the requirement to identify other-contractor related ECPs in Section I, makes it feasible to realize two aspects of the apparent Figure 13 intent, namely: to track ECP implementation status both across impact documents and across contractors.

5.4 VERSION DESCRIPTION DOCUMENT (VDD)

The version description document is a document prepared to accompany the delivery of a CPCI or of CPCI changes. It identifies the items delivered and records additional data relating to the CPCI status and usage. Its functions are:

- To provide field personnel with necessary information and instructions pertaining to the delivered version or interim change; and
- To provide configuration management with a record that permits identifying the exact configuration of the CPCI which is approved for use at the time of delivery.

5.4.1 Definitions and General Policy

A version is the actual configuration of a CPCI which is introduced at a given time for installation and test or operation into the system in the form of a magnetic tape, disc, card deck, or other. A new version is created: (a) when a newly developed item is prepared for its first formal delivery; or (b) whenever the CPCI is completely reassembled to contain all Class I and Class II changes accomplished since the preceding version.

An interim version occurs when a Class I change is introduced into an existing version through delivery of partial changes to the code, short of complete reassembly and delivery of a new tape or card deck.

Versions and interim versions are prepared by the developer (or, later by the responsible computer programming support center for the system) in the form of a master tape/deck from which duplicates are made for delivery to test or operating locations. In some systems, capabilities also frequently exist at each site to make further duplicates and to alter the configuration for various test or operating purposes. However, those alterations do not constitute new versions or interim versions; the latter are issued only by the developer (or other center), where configuration management functions are maintained centrally for the system. Certain aspects of that situation pertaining to the system DT&E site are discussed further in the next section (Section 6). In general it should be noted that:

- The controls in effect at a test or operating location are local controls for which a test director or site commander is likely to have primary responsibility. Although provisions are normally made for effective interaction with centralized technical, data, and configuration management functions, the local Air Force controls are likely to be matters over which the developer's configuration manager, for example, has little or no jurisdiction.
- AFSCM/AF LCM 375-7 contains the general provision that Class II changes may be incorporated into the CPCI as they occur. Such alterations do not constitute new versions or interim versions requiring the preparation of a VDL, although each new version/interim version issued to incorporate one or more Class I changes must also include all Class II changes that have occurred since the preceding version.
- Strictly speaking, "Class II changes" incorporated at a field location and then reported to the computer programming center (developer) should be regarded as authorized deviations, rather than as changes, until such time as they become incorporated into approved SCNs to the specification. Depending on circumstances, they may have to be altered to reconcile discrepancies among sites, or may be outdated by upcoming Class I changes to the affected portions of code, before being formally processed as changes.

5.4.2 Numbering Versions and VDDs

Versions of a CPCI are numbered consecutively, normally beginning with "1" for the version delivered for audit at PCA. Interim versions are identified by attaching a letter to the number of the current version, in alphabetical sequence for successive interim versions; for example, Version 3B represents the second interim change to the third complete version of the CPCI.

The number of the VDD corresponds with the number of the version or interim version which it accompanies, but preceded by "VDD-"; for example, the VDD for Version 1A is VDD-1A.

5.4.3 Preparation and Content

Additional requirements for identification data to be contained on the title page of a VDD are set forth in paragraph 80.12.1.1 of MIL-STD-483. Instructions provided for the VDD contents (para. 80.12.1.2) are relatively clear as written, covering the ten sections listed and summarized briefly here in Figure 5-10. Note that:

VERSION DESCRIPTION DOCUMENT	
a. INVENTORY OF MATERIALS RELEASED	[List of all items (tapes, cards, discs) covered by the VDD, by CPCI and version number. All related release documents for support items required to operate, load, or regenerate the released CPCI.]
b. INVENTORY OF CPCI CONTENTS	[List of all computer program instructions and data content released.]
c. CLASS II CHANGES INSTALLED	[Number, title, and issue date of each Class II CR; related SCN numbers and issue dates.]
d. CLASS I CHANGES INSTALLED	[Number, title, and issue date of each ECP; related SCN numbers and issue dates.]
e. ADAPTATION DATA	[When applicable: Identification of all unique-to-site (or mission) data contained in the item released.]
f. INTERFACE COMPATIBILITY	[Identification of other systems/CIs/CPCIs affected by incorporated changes, and present status.]
g. BIBLIOGRAPHY OF REFERENCE DOCUMENTS	[Listing of all pertinent documentation.]
h. OPERATIONAL DESCRIPTION	[Operational effects of Class I and II changes incorporated.]
i. INSTALLATION INSTRUCTIONS	[Methods to install and checkout the delivered version.]
j. POSSIBLE PROBLEMS AND KNOWN ERRORS	[Needs for further testing; status of problem resolution.]

Figure 5-10. Version Description Document: Summary of Contents

- The information pertaining to Class II and Class I changes (sections c and d) is not required for the first VDD (VDD-1).
- The adaptation data information (section e) applies only to those CPCIs for which a part of the fixed data base consists of data values that vary among individual site locations, or perhaps for different missions. The configurations at individual sites are normally identified as types within a single CPI (see 2.3). Depending on the system, changes to the adaptation data may be incorporated into a new version either prior to delivery or at the time of field installation.
- The bibliography of reference documents (section q) is not required for interim versions.

SECTION 6. CONTROL DURING SYSTEM DT&E

Preceding sections have addressed configuration management requirements and procedures as they apply during development, and as they involve relationships which are largely confined to the developer and procuring activity. Beyond the time of PCA, additional factors come into play which must be taken into account in the planning and management of each program, but for which guidance provided in the standards is relatively sparse. One reason for that sparsity is the fact that circumstances vary widely in different programs with respect to such factors as responsibilities, locations, and initial conditions. This section describes how configuration management has been carried out during the system test period in a limited number of past system programs, in order to illustrate the nature of questions that can be encountered. Assumptions are identified which can affect how, or whether, the practices described may be relevant to other programs.

The existence of a system test location, together with a test organization which is responsible for controlling and conducting the test operations, is the primary source of additional factors to be taken into account. That expanded situation is illustrated in summary outline in Figure 6-1. It is also a potential prototype for the operational phase, in that one or more operational sites may have relationships to a centralized CCB and computer program support agency similar to those described below for the system test site.

6.1 ASSUMED INITIAL CONDITIONS

The period in question is the period of system DT&E, which begins following installation and checkout and continues until about the point of system turn-over and transfer. For purposes of this description, the basic assumption is made that practices described in preceding sections of this guidebook have been implemented, and that the development of the major mission CPCI(s) has been accomplished with reasonable success. Additionally:

- The original developer remains on contract through the system DT&E period, in part to perform on-site support to the system test activity.
- CPCI qualification has been completed, with the possible exception of a few requirements which can be verified only in the full system environment.

- The Part II specification has been completed and verified for accuracy/completeness as the as-built description of the (conditionally-qualified) CPCI. PCA has been accomplished, but:
 - a. The developer continues to be responsible for completing qualification, possibly via subsequent formal qualification review (FQR).

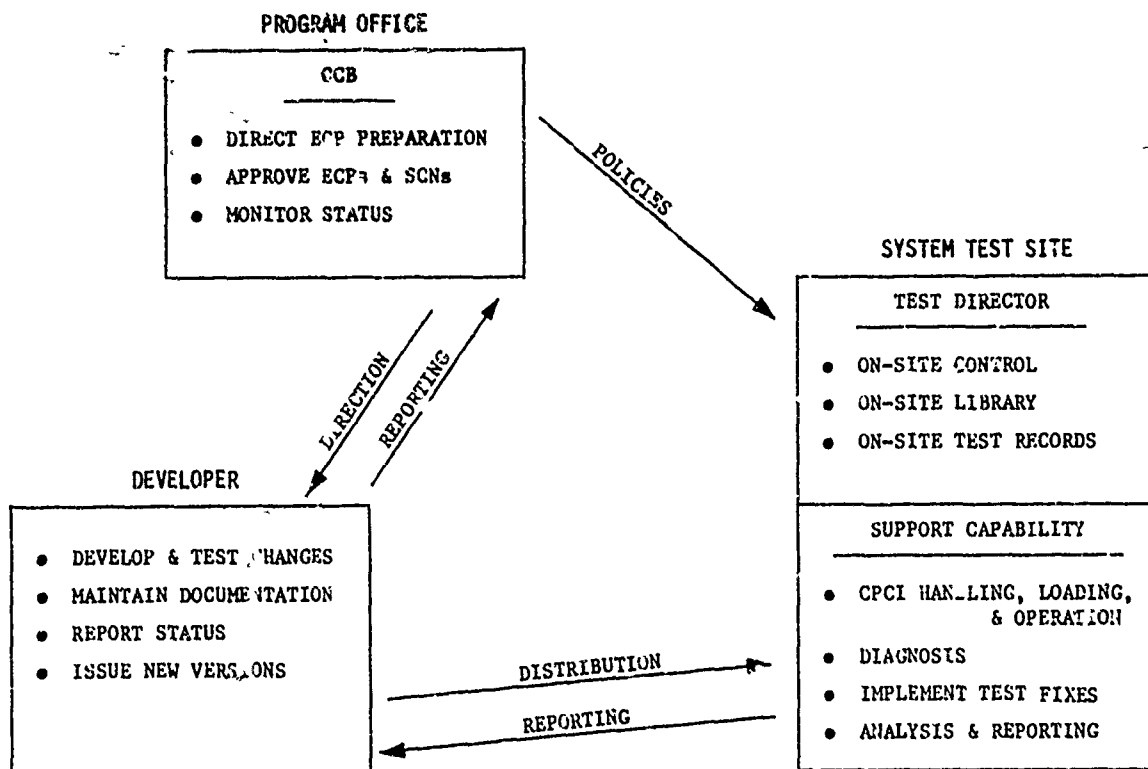


Figure 6-1. Relationships Among Activities During System DT&E

- b. The developer may also be responsible for, and in the process of, implementing ECPs for new requirements not incorporated in the initial CPCI version.
- Formal configuration control is maintained by the program office CCB; and the developer continues to implement configuration management procedures at his home plant to maintain normal control and status reporting. Those procedures include:
 - a. Formal processing and/or reporting of Class I and Class II changes.

- b. Specification maintenance to reflect all changes.
- c. Maintenance of related documents to reflect impact of approved changes to the CPCI/specification.
- d. Periodic reporting of status for ECPs and maintainable documents.
- e. Delivery of new, CCB-approved CPCI versions and accompanying VDDs in accordance with established criteria.

6.2 ON-SITE CONTROL AND SUPPORT

On-site control is exercised by the designated authority at the test location, i.e., the test director. Although the test director is likely to be a member of the program office CCB, his activities in that on-site role are separate from those of the CCB as such. His on-site controls should be subject to verification by the CCB; however, he should have the authority and capabilities to take advantage of the inherent flexibility of software to support the test operations, within reasonable limits. For example, he should be able to authorize test deviations* needed to keep the CPCI in operating order as the testing progresses, or to create a desired new test condition, or to evaluate temporary fixes for difficulties encountered.

Thus, the CPCI in actual use at the site may consist of one or more "test versions". A test version is an altered copy of the current approved version. As the testing progresses, additional copies may be made to contain successive test alterations and are identified by supplementary letters, numbers, dates, and/or times to permit linking the CPCI configuration actually used with individual test operations.

The test director is responsible for instituting measures to control the handling and storage of CPCI test versions and their operation in the computer -including, for example, measures to assure that only authorized fixes are inserted and that records are kept to permit verifying the exact configuration of the CPCI at the time of each test.

Technical computer programming support is available on-site to perform such functions as:

*The term "deviation" is used here in a general sense, referring to a departure of the item configuration from its approved specification. A deviation becomes a change when a corresponding change is made to the specification.

- Operating Support. Handling, loading, and operation of the CPCI.
- Trouble-Shooting. Diagnosis of malfunctions or undesirable CPCI or system performance.
- Fixing. Designing, coding, and inserting error corrections or other alterations for test and evaluation.
- Analysis and Reporting. Based on the results of test and evaluation, formulating recommendations and reporting to the developer's configuration manager:
 - a. Class II Changes - e.g., error corrections made and to be retained in the on-site version, reported via draft (recommended) Class II CRs to the home plant.
 - b. Class I Changes - preparation of the basic technical content of ECPs proposing significant redesigns to be incorporated in a new version or interim version of the CPCI when processed by the developer and approved by the program office CCB.
 - c. Problems or deficiencies requiring study, analysis, or implementing capabilities exceeding the on-site resources.

6.3 SUMMARY CHARACTERISTICS

In that augmented working environment, the complications are ones which affect principally the technical and test activities. Reports of problems and change proposals received from the field are first screened by system and software engineering personnel at the developer's home plant before formal processing of changes is initiated; the processing of changes then continues to occur as described above in Sections 4 and 5. If ECPs previously directed by the CCB are in process of being implemented, draft Class I or Class II changes input from the field must be reconciled with those before being converted into formal ECPs or CRs for submittal to the program office. For example, the affected portions of code may be undergoing a complete redesign in the upcoming new version of the CPCI.

The developer's configuration management activity is necessarily expanded to keep track of deficiency reports and draft change proposals input from the field, to assure their proper disposition. Special forms and processing

procedures for handling the field inputs have been worked out and used extensively in a number of individual system programs; but uniform requirements in that area have not yet found their way into the DoD/Air Force standards for general use--again, largely because the circumstances and organizational relationships tend to vary widely.

Whatever those complications, however, their principal effect on the developer's configuration manager is to introduce additional sources of original requirements leading to the initiation of proposals or reports of Class I and Class II changes. His principal concern continues to be with centralized control of the CPCI specification and the related status keeping/reporting procedures described in earlier sections of this guidebook.

SECTION 7. NOTES

This section is devoted to a few notes which respond to selected questions raised by reviewers of this guidebook in its draft form. The notes refer to separate topics covered in various earlier sections, and are not necessarily related. The few notes collected together to form this special section are ones which are either too lengthy to be inserted elsewhere in the form of footnotes, or whose content tends to be peripheral to the orientation and emphasis of those topics as they are presented in the basic text.

7.1 COMPUTER PROGRAMS AS DATA vs. CIs

In discussing considerations which led to computer programs being classified as configuration items for purposes of acquisition management, the statement is made in the text that a computer program is intrinsically an item of data (1.4). Elsewhere in the text, various differences are identified in objectives and appropriate procedures for managing computer programs as compared with equipment CIs. This note is written to further interrelate those two points, and to suggest that they can provide, jointly, an improved insight into a number of prevalent questions and problems.

"Data", in this particular context, refers to reports, forms, manuals, specifications, and other items of the classes which are acquired via CDRLs. Data management practices in DoD recognize that those are not confined to items written or printed on paper; they include any information recorded in suitable form on any suitable medium, such as film, photographic paper, magnetic paper or tape, and in digital or analog form.

When the practice of using CDRLs was initiated at ESD (1964), a number of newly-appointed data managers using the form as a retrofit to on-going programs included computer programs as prominent items of data in their first listings, before learning that the practice was inconsistent with AFSC's recent (at that time) decision to manage computer programs as configuration items. That particular "misunderstanding" remained corrected until late 1974, when a revision of the ASPR appeared requiring that "computer software" be listed on CDRLs as a measure to protect the Government's rights in data (Defense Procurement Circular 74-3, November 1974). Recognizing the inconsistency, AFSC initiated an attempt to have the ASPR committee reverse its decision, but at the same time developed the current workaround procedure of requiring computer programs to be listed in both the contract schedule (as for hardware deliverables) and the CDRL.

As regards data rights, the ASPR appears to have a basis in legal decisions which have been reached in determining whether computer programs are things to be copyrighted and/or patented. The rulings that have been made tend to favor treating computer programs more like data than like hardware.

In the context of system acquisition management, data items are distinguished from equipment items partly because of intrinsic differences in their physical nature, but more directly because of derived differences in the typical requirements for realistic and effective management of their acquisition and support. As examples:

- Most data items can be adequately specified by a generalized DID, together with a few specific instructions, whereas an adequate specification for an equipment item typically involves an extended process of analysis, design, fabrication, and testing.
- Many data items are required as basically only "one-of-a-kind". But if a given data item is needed in quantity, it can be printed or otherwise duplicated on a general-purpose machine (e.g., a printing press). Equipment items are normally procured in quantity; and the "duplication" of units requires a special-purpose, normally costly, manufacturing capability.
- Considerations in such areas as reliability and maintainability are normally significant for equipment items, since equipment typically degrades through use and factors of environment. While recording media can also wear out, they are relatively easy to replace; and the substantive information contents of a data item are not subject to the same factors of undesirable change.
- Equipment items in military systems typically require provisions for their operation in the field, e.g., in the form of fuel, electrical power, or ammunition. No similar requirements exist for data items.

That list of differences could obviously be expanded, and could also include many ways in which indicated approaches to managing equipment and data items are similar--independently of any considerations specific to computer programs. The reasons outlined in the text for classifying computer programs as configuration items (1.4) are that some significant management requirements for computer programs are more like those typically suited to equipment than to data items, with respect to selected requirements which are normally different for those two classes of items. But it should be noted that the comparison did not extend to cover many ways in which the reverse decision might be indicated. Among the few comparisons listed above, for example, it may be observed that computer programs share much more in common with data than with equipment.

Although account was clearly taken of the above points in the course of AFSC's initial development of policy for configuration management of computer programs,* many of their implications for related areas of acquisition management have not been further explored and documented. In the absence of more positive guidance to the contrary, the decision to manage computer programs as configuration items is often assumed to be synonymous with the decision to manage them as equipment. There are many ways, however, in which POs may find the opposite assumption to be more productive. The tendency of computer programs to exhibit their basic character as data can be discerned, for example, in some of the differences in specification roles outlined previously, in 3.6 of the text. More generally: Attempts to apply procedures based on established hardware practice in such areas as production, logistic support, maintainability, and reliability typically lead to confusion, debates, and/or misunderstandings. Much of the confusion tends to disappear when it is fully understood that those concepts apply (rather, fail to apply) to software in essentially the same manner that they would to a technical manual, and for the same reasons.

7.2 "SOFTWARE" AND THE DoD DIRECTIVE

Among many examples of the "jargon and mystique" which have been said to characterize sectors of the software community for the past two decades, the term "software" itself is perhaps one of the best known. Like others, it is a term borrowed from outside of that context, defined to suit the initial borrower's purpose, disseminated widely--and frequently redefined to meet other purposes of new users. Definitions which the author has encountered (some formal, some implied by use), include the following:

- Deliverable contractor data, such as handbooks, manuals, formal reports, and engineering drawings--as opposed to deliverable hardware. This is probably the original use of the term, adopted by aerospace industries in the early 1950s. It is still widely used with that meaning. A curious carryover to the environment in which computer programs are also being managed basically as configuration items (like hardware) occurs in the current Appendix F to AFR 65-3, in the statement, "...software associated with computer programs will be managed in accordance with AFR 8-2...".
- Special support computer programs developed by a computer manufacturer and provided with sale and delivery of the computer. This is probably its first use in the ADP community--the "door opener".

*See the discussion in paragraph 1-36, "Computer Program vs. Equipment CIs", of AFSCM/AFLCM 375-7.

- All computer programs (the use adopted in this guidebook, to conform with the title and intent of this guidebook series as a whole).
- Computer programs plus their documentation (e.g., in the Army's MIL-S-52779).
- All computer programs, plus all products associated therewith, including documentation and the computer itself.
- All efforts and products supplied by a computer programming contractor, including deliverable documentation, training, support, and other services in addition to the computer programs.

7.2.1 Air Force Practice

Most of the coverage specific to computer programs contained in current Air Force standards derives from a project which was initiated by ESD in 1964 and directed by a special committee formed for the purpose. Although the committee started with the title, "ESD Software Management Committee", it undertook a study of that term as its first task; and the result was to bar its further use in the project. One significant longer-term effect of that decision is that "software" still does not appear in such current documents as AFSCM/AFLCM 375-7, MIL-STD-483, Air Force DIDs for computer program data items, AFR 65-3 (except for the anomalous use noted above, in Appendix F), or in either volume of AFR 800-14.

It is of some interest that the ESD committee took that course rather than attempting to construct its own definition--and to their credit, in the author's opinion, since the ability of any one Government agency to have real success in overcoming a diversity that well entrenched is inherently limited. In contrast, there was no handicap of similar magnitude attached to the use of the term "computer program". In defining the latter, the following two points received attention:

- To avoid confusion, a computer program is not referred to in official documents as simply a "program", since "program" normally refers to the system program, in the context of Air Force system acquisitions.
- A computer program consists basically of computer instructions, but also includes those data values which are coded and contained in the item at the time of its delivery. This point is not only consistent with generally accepted practice; it is a significant aspect of the definition for purposes of acquisition and control.

Those points are reflected in the Air Force standards, including the Joint Services regulation (AFR 65-3), and in explanations/definitions provided in this guidebook (1.6.1 and the glossary).

7.2.2 Recent Elements of Confusion

The present DoD Directive 5000.29 (April 1976) uses and defines the related terms: computer software, computer data, computer program, and software engineering. It has been brought to the author's attention that those definitions can be interpreted to be in conflict with the definitions provided in this guidebook for both "software" and "computer program". A careful reading of the directive confirms that: (a) both the definitions and uses of those terms in DoDD 5000.29 are indeed sufficiently loose that their real intent is ambiguous in a number of respects; and (b) they may well prove to be in significant conflict with established Air Force practice. Specifically:

- "Computer software" is defined as a combination of computer programs and computer data.
- "Computer program" is defined in its normally accepted meaning, including familiar examples, except that coded data values are not explicitly identified as being a part of the content, which consists of "instructions or statements".
- "Computer data" is defined as "basic elements of information used by computer equipment in responding to a computer program".
- "Software engineering" is defined, in essence, as engineering of computer software.

Thus, the confusion introduced by those words consists jointly of some basic ambiguity and some potential conflicts with widespread practice. Summarized very briefly, those include the following:

- The interpretation can clearly be made that: a computer program consists solely of the computer instructions; all data involved in the computer program operation are classified separately as computer data; and computer software is the combination of computer programs and computer data. The further interpretation can be made that "computer software" directly replaces the term "computer program(s)" for purposes of acquisition management in defense systems. (If this interpretation is really intended, and were to be officially accepted by the Air Force, it would impact not only

this guidebook but also the spectrum of current Air Force standards governing acquisition management of computer programs; it could be debated whether such an extensive and time-consuming change would even be feasible to accomplish.)

- In the absence of positive statements in DoDD 5000.29 to the contrary, however, it can also be argued that the terms "computer program", "computer software", and simply "software" are all really intended to be interchangeable; for examples, note the directive's uses of those terms in paragraphs V,C, V,D,3, V,E, V,F, and V,G.
- The intent with respect to "computer data" is obscure by virtue of both its brief definition and the absence of references to it in the directive's content. One possible interpretation is that it refers only to live inputs, as opposed to data values coded and inserted into a computer program prior to its operation; others are also possible.* The directive's expressed purpose is to spell out policy for the acquisition management of computer resources in defense systems. Coded data associated with computer programs pose certain real questions, since: some can be included in a computer program at the time of its delivery; some can be input for processing during operation; and still others can be procured separately (see 1.6.1). But those distinctions and their implications for acquisition management policy are not addressed.

Hence, it seems clear that various interpretations of those terms can be made and defended. With regard to the "real" intent of the people who formulated and coordinated the directive (involving numerous inevitable compromises on points of issue), one can speculate that "software" may have been deliberately chosen to serve its traditional function of suggesting whatever each affected reader might wish to believe.

7.2.3 Summary

To the author's knowledge, no Air Force action has yet been taken to rule on the applicability of DoDD 5000.29 to Air Force procurements. Until such action may be taken, provisions of the directive which are not presently covered in Air Force regulations or other documents do not legally affect Air Force activities. It should be noted that the substantive policies of DoDD 5000.29 were anticipated and are already covered in AFR 800-14. Since the definitions outlined above could have serious impact on Air Force practice (again, depending on interpretations), it is to be expected that any ruling on their specific applicability will be preceded by careful study.

*For example, see p. 171 of ESD-TR-76-159 (Schoeffel, W.L.).

In the interim, the definition of a computer program provided in the current Air Force and DoD standards for configuration management is the only definition which can reasonably be accepted and used in this guidebook. That definition will continue to govern both Air Force PO's and their contractors until such time as it may be changed in the standards, specifically, and further reflected in the specified requirements of individual system contracts.

7.3 SYSTEM SEGMENTS

The purpose of this note is to provide a few comments on the meaning and uses of "system segment", to supplement the brief treatment of this topic made in 3.1.1 of the text. Like the configuration management standards, this guidebook confines its coverage of that topic to its implications for configuration management procedures. However, questions have been raised about the system segment concept with respect to its system engineering and procurement aspects, for which corresponding coverage in current regulations, specifications, and standards is relatively sparse.

A system segment is defined as a discrete package of system requirements for which responsibility is assigned to one contractor or Government agency (see 9.1). Instructions for preparing a system specification provided in Appendix I of MIL-STD-490 do not include any reference to system segments, but use the term "functional area" instead. As described in 3.1.1 of the text herein, Appendix III of MIL-STD-483 provides the instruction to substitute system segments for functional areas in the general volume of the system specification, in the special case where it has been decided to prepare separate volumes for individual segments.

To the author's knowledge, however, direct answers are not provided in any of the current standards to such questions as: whether system segments should be substituted for functional areas when the system specification is prepared as a single volume; whether there is any difference between a system segment and functional area other than the label; and whether, when system segments are identified, they are necessarily assigned to separate contractors or Government agencies. Some of the known considerations which can be brought to bear on those questions, together with some of the author's opinions, are summarized briefly as follows:

- The system segment concept is derived from the concept of subsystems, which in turn is based on the normal need to break down a complex system into a next-lower level of assembly before reaching the highest level which is appropriate for breaking out individual configuration items of hardware and software (i.e., items of "defense materiel", as opposed to people and

other system elements). The original purpose of the system segment concept was to impose on system engineers responsible for generating system design the requirement to take into account program management as well as technical considerations in arriving at that first-level breakout. Specifically, it introduces the requirement that each major piece be defined in such a way that responsibility for its development can be assigned to a single organization.

- It has been observed that the management principle involved is basic to more than just that first level. It is also reflected explicitly at the next-lower level, through the requirement that a CI be technically identified as something which can be assigned to a single organization. In an orderly management scheme, the same principle will be extended to successively-lower levels, perhaps down to the point at which responsibility for the smallest piece can be assigned to one person. The general objective is to achieve a structure of technical design (generation breakdown; see 3.1.4) which can be readily correlated with contracts, specifications, work tasks, organizations and organizational levels, technical documentation, supervision, budgets and cost accounts, etc., from top to bottom.
- Some brief history is relevant to this topic. Appendix I of MIL-STD-490 is based directly on Exhibit I of the former AFSC Manual 375-1.* The most noticeable differences between the two are that (a) some of the Air Force terms were changed in the course of DoD coordination, and (b) much of the explanatory guidance disappeared--including the wealth of associated guidance in its companion manual for system engineering management, AFSCM 375-5. With regard to the point in question: The 1964 issue of AFSCM 375-1 introduced the term "system segment" as a replacement for "subsystem", explaining that the two terms were basically interchangeable if a subsystem is defined with a view to its organizational implications. In preparing MIL-STD-490, the DoD committee arrived at the term "functional area" as a direct replacement for "system segment".
- Requirements set forth jointly in MIL-STD-490 and MIL-STD-483 for specifying functional areas are effectively the same as they were previously for system segments in such significant areas as: allocations of system functions, identification and definitions of inter-functional area interfaces, and identification of CIs contained in each functional area. Thus, technically, the only difference between functional areas and system segments is in the label itself.

*The original issue of this manual, in 1962, introduced the term "configuration management" and the concept of uniform specifications (its full title is provided in the earlier footnote, basic paragraph of Section 2). The issue being referred to here is: AFSCM 375-1, "Configuration Management During Definition and Acquisition Phases", 1 June 1964.

- While it is not always readily apparent from the Air Force definition alone, there has never been a policy that system segments are necessarily assigned to separate contractors --only that they be identified in such a way that they can be assigned separately. The approach to contracting for each system acquisition is based on other criteria. Multiple segments can be (and have been, perhaps more often than not) assigned to a single prime or associate contractor.
- In the absence of any clear policy statements to the contrary, POs are free to identify either functional areas or system segments, or both. The only restriction which appears to exist is that if system segments are identified, attention must be paid to their implications for acquisition management. In view of considerations outlined above, continued observance of that rule would appear to be the appropriate course for POs to follow, independently of the label chosen.

7.4 PROBLEMS WITH CHAPTER 2 OF AFR 800-14

Reviewers have noted that the description of the development and control process for the system specification, provided in 3.2.2 of the text, is discrepant with statements made in Chapter 2 of AFR 800-14, Volume II. Specifically, that chapter states (para. 2-3) that "the initial system specification" is a product of the conceptual phase, whereas (para. 2-4) the "authenticated system specification" is a major product of the validation phase. Taken together, the statements indicate that authentication (hence, baselining for configuration control) does not apply to the initial system specification, but only to the specification in its completed form at the end of validation.

The basis for those statements is not known. They disagree with established practice and Air Force policies stated elsewhere, as well as with the description given in 3.2.2 herein. As examples, the following statements are to be found in paragraphs 3-5 and 3-7 of AFSCM/AFLCM 375-1 (March 1971):

"When the system specification has been developed to the extent required to define the Air Force functional requirements for the system, it is authenticated by the SPD and made directive, in contracted validation phase contracts for all contractors, as the technical performance base for the system program. The system specification defines the approved system functional baseline... The authenticated system specification will be included in the request for proposal during the validation phase... Should the Air Force decide to change or add to the system specification after the initial issue has been authenticated, the normal ECP procedure... will be followed. This is essential in order that persons concerned with the program, in both Government and industry, may be kept informed of the exact content of system requirements.

"...Industry proposals for the validation phase contract and the subsequent design and development contract which they prepare and submit during the validation phase should be responsive to requirements for engineering control to the negotiated system functional baseline (system specification). ...All proposed changes to the system specification will normally be accumulated by the contractor during the contracted validation effort and prepared as a single ECP package for presentation to the Air Force at the end of the contractor's validation effort."

Further indication that those precepts have not materially changed in recent years are provided in paragraph 2-21,c of AFSCP 800-3 (April 1976), as follows:

"Functional Baseline. The functional baseline (program requirements baseline) is established by the end of the Conceptual Phase. It includes broad system performance objectives (in the format of a MIL-STD-490 Type A specification),... the system specification defines the technical portion of the program requirements baseline. The Air Force and OSD use this information to evaluate the proposed program and to compare it with competing programs. After review and approval, this baseline is the basis for the Validation Phase."

The points to be derived from those sources which are generally basic to established Air Force practice are:

- The initial system specification is authenticated and baselined, prior to initiating validation phase contracts.
- The initial system specification defines the system functional baseline, beginning with its authentication at the outset of a validation phase. The system functional baseline continues to be defined by the initial system specification plus accrued SCNs resulting from approved ECPs.
- The expanded system specification, at the end of validation, results from the ECP package generated during the validation phase.

Further study of that particular chapter of AFR 800-14 (Chapter 2) reveals that its content is also discrepant in many other ways with accepted practice and the source documents. For example:

- Paragraph 2-5,a states, in lines 6-11, that PDR is held "to review the preliminary design against the respective authenticated development specification" (a correct statement). Yet, at the very end of that same paragraph, it states that preliminary design information (described very loosely in the preceding sentence) "should be contained in the development specification and become the basis for PDR of the computer program." (!) The accompanying diagram, Figure 2-1, also portrays that latter, common misconception. (The actual purpose of PDR is not to critique the development specification; it is to review the design approach proposed to meet requirements of the previously-authenticated development specification.)
- Interfaces are supposedly "finalized" at CDR, including interfaces with personnel (!) (para. 2-5,b). External interfaces for CPCIs are functional, and should have been finalized prior to PDR; internal interfaces are not likely to be finalized until coding and testing have been accomplished; human performance requirements are not managed as "interfaces".
- Formal test plans are "initially submitted in preliminary draft form for review at CDR" (para. 2-5,c). CDR is far too late; initial submission of the test plan should occur in the validation phase (see Block 7 of DI-T-3703), and the updated test plan should be completed shortly after PDR.
- Satisfactory formal testing of a mission CPI may not be completed until "completion of... OT&E" (para. 2-5,c). A more meaningful statement, in the acquisition management context, would be that satisfactory qualification of the CPI may not be accomplished until system DT&E. Qualification has to be accomplished prior to turnover.

Those discrepancies can only be interpreted as errors, since they are inconsistent internally with other content of AFR 800-14 itself as well as with the source documents which AFR 800-14 references freely. They suggest that Chapter 2 may have failed to receive adequate review and coordination prior to being incorporated into the regulation at the time of its final issue.

7.5 CONFIGURATION INDEX - QUESTIONS

This note is written to provide a further discussion of questions that have been raised regarding the specific nature of conflicts contained in the MIL-STD-433 instructions for the configuration index. Its purpose is to record some background factors associated with the treatment provided in MIL-STD-483, and to clarify reasons for the approach taken to this topic in the text (5.2).

A few of the pertinent background facts and circumstances are summarized very briefly as follows:

- The basic concepts and content of the configuration index, change status report, and version description document were developed during the 1950s for control and status reporting of computer programs in the SAGE system, prior to the time that "configuration management" became recognized (initially for system equipment) as an acquisition management discipline.
- Descriptions of the three documents were first disseminated for general use in the ESD Exhibit EST-1 (1966), which was prepared as a computer program supplement to the 1964 issue of AFSCM 375-1. Those instructions were based most directly on the manner in which the documents were actually being prepared and used at that time in the BUIC III acquisition. (However, to avoid incorporating features which might be peculiar to BUIC III, and to simplify their explanation, the generalized descriptions in the ESD exhibit were modified in minor ways. For example, the configuration index and change status report were described basically as they should appear if a contractor happens to be responsible for only one CPCI--although for both reports, the actual practice in BUIC II, BUIC III, and SEEK DAWN/818 was to prepare one report per contractor, covering all CPCIs for which each contractor was responsible.)
- While the basic concept of the index as a periodic report of document status is relatively simple, once understood, its description can prove to be awkward, particularly when limited to a bare statement of minimum requirements. The initial description in ESD Exhibit EST-1, although considered meaningful and obvious to its authors, proved to be confusing to people who had not previously worked with the report--even with the simplifying assumption of a single CPCI. Neither at that time nor later was any supporting guidance prepared and disseminated to provide samples, clarify objectives, or discuss alternatives appropriate to varied circumstances.
- The effort of updating and rearranging the major content of ESD Exhibit EST-1 for incorporation into AFSCM/AFLCM 375-7 and MIL-STD-483 (1969-71) was carried out largely by a small task force of people who had not had experience with those computer program status reporting documents, but--as is still characteristically true for configuration management standards--had extensive backgrounds in configuration management practices and principles for systems/equipment. In addition, they were working under pressures of limited time, and with limited support.

In the course of that last event, the instructions for the index managed to retain all of the earlier elements upon which the description provided in 5.2 above is based, but at the same time acquired certain new elements. One new element was Section A, the CPCI historical record, which serves understandable functions and has not occasioned any problems, to the author's knowledge. Ignoring for the moment the existence of the MIL-STD-483 Figure 13 and associated instructions in paragraph 80.10.4.1, a careful study of the basic paragraph 80.10 and its first subparagraph, 80.10.1, will reveal that they still provide a direct basis for preparing the Part I of each major section in the manner described in this guidebook. (It is not being suggested that those instructions by themselves are adequate, except for readers who may recognize their source, original intent, and/or correspondence with the treatment in 5.2 herein.)

The problems arise in understanding and reconciling all of the MIL-STD-483 instructions for the index, including Figure 13 and its accompanying instructions in paragraph 80.10.4.1.* The conflicts become most evident if one actually attempts that exercise. But, as examples:

- The title of Figure 13 is "Configuration Item Development Record--Section I, Parts I and II". Why is this called a development record, which is the title of Section A but not of the entire index? Why is the use of Arabic and Roman numerals reversed here as compared with the text?
- If this model (noting the Figure 13 content) applies to Sections I and II, and lists all documents affected by each ECP: What does one do in Sections III through VI? In fact, what does one do about the "other documents" in Section II, including the other specification part (I or II) in each of those two sections--duplicate them?
- What are the mechanisms by which an integrating contractor obtains this information from separate associate contractors, when those exist? What happens to the format when they do not exist?
- Is the listing for the Part I specification initiated in Section I of the index issue following delivery of its SCN/ECP, or is it withheld until updates of all impact documents, including related SCNs/ECPs within and/or across contractors are issued? Is "date of issue" (called for in 80.10) reported only for the basic issue, and not for any of the SCNs? What is there to indicate whether the listing which appears in a given issue is current, or complete with respect to impacted other documents and contractors?

*The first sentence of 80.10.4.1 references "Figure A" instead of Figure 13; however, that error is incidental.

- c What are the intended objectives of information to be provided in this form; who is (are) the primary intended user(s)?

The author is indebted to Mr. Charles Bashaw, ESD/DRT, for recently shedding light on the origin and probable intent of those instructions, and for suggesting some related considerations outlined below.

It was mentioned above that the description of the configuration index provided in ESD Exhibit EST-1 was only marginally adequate to convey an understanding of its content, preparation, and functions to people who had not actually worked with it. In the course of revising that description for inclusion in MIL-STD-483, it was decided to clarify (and perhaps simplify) it by adding requirements for specification maintenance records similar to those set forth in Appendix VII for equipment specifications. If one compares the descriptions and figures provided now in both Appendix VII and VIII, it is fairly obvious that Section A of the configuration index is the direct counterpart of the equipment CI development record, Part 1 (Figures 12 and 8, respectively). However, a comparison of Figure 13 with Figure 9 indicates that those two are also the same, basically, except that Figure 13 (a) adds the requirement for reporting other documents and (b) includes a sample of Part 2 data for the section. Thus, briefly summarized, Figure 13 clearly represents an attempted combination of the equipment CI development record, Part 2, with the computer program configuration index--albeit, a combination in which the elements have proved to be somewhat incompatible.

In the light of that interpretation, it is pertinent to inquire whether the computer program status reports provide information equivalent to that provided in the equipment specification maintenance records, if needed. The brief answer to that question is that: All were required in the former ESD Exhibit EST-1, and are currently required by Appendix VIII of MIL-STD-483 (independently of paragraph 80.10.4.1 and Figure 13), except for the record of impact on or by related ECPs across contractors. The author's suggested addition to the change status report (5.3.1) provides that one missing element, if and when a multiple-contractor structure makes it applicable.

However, it appears that one function of the CI development record, Part 2, is to record the history of SCNs/ECPs in a form suitable to accompany each specification at the time responsibility for its maintenance is transferred to the supporting command. While no similar requirement is known to exist for computer programs (many of which ESD has traditionally transferred to the using commands), it is conceivable that it might be encountered. If it were, and in just that form (i.e., for specifications only, separately for each CPCI, and including impact on specifications by related ECPs), the information would have to be extracted from the configuration index and change status report, jointly, and reformatted.

At the same time, coverage provided by the computer program status reports is much broader than specification maintenance. In combination with the version description document, they also effectively accomplish, for computer programs, functions generally analogous to those of post-PCA configuration status accounting reports for equipment. The fact that they do it prior to PCA, as well as after, has suggested to some observers that similar coverage is really needed for equipment--i.e., during the initial development period. In a few cases, contractors have been required to issue the change status report as a single monthly report covering all of each contractor's ECPs, whether affecting equipment CIs, CPCIs, or both. (The particular examples of those which the author has seen actually placed their predominant emphasis on ECPs to equipment CIs, to the extent of omitting significant elements of the required status information for ECPs to computer programs.)

One original purpose of the configuration index is to ensure that key documentation associated with computer programs is developed and regularly updated to reflect the current configuration of each item. Failures to accomplish that purpose have been a traditional and pervasive source of software user troubles and expense. As outlined in the guidebook, the configuration index and change status report are reports which should be prepared by the developer--and by each developer, if the program involves more than one. The information is useful not only for its nominal purposes, but can also be invaluable to a PO in providing indicators of how effectively the contractor's configuration management is integrated with his total CPC development effort, throughout the acquisition (see 5.2, basic paragraph, also Figure 4-5 and 4-6). In short, they can function as significant techniques in the PO's overall approach to acquisition management of computer programs, when sufficient emphasis is devoted to ensuring their proper preparation and uses.

SECTION 8. BIBLIOGRAPHY

8.1 REGULATIONS, SPECIFICATIONS, AND STANDARDS

- DoDD 5000.29 Management of Computer Resources in Major Defense Systems.
26 April 1976.
- AFR 57-4 Retrofit Configuration Changes. January 1972.
- AFR 65-3 Configuration Management (Joint Services Regulation)
1 July 1974.
- AFR 800-3 Engineering for Defense Systems. 30 August 1973.
- AFR 800-14, V. I Management of Computer Resources in Systems.
10 May 1974.
- AFR 800-14, V. II Acquisition and Support of Computer Resources in Systems.
26 September 1975.
- AFLC Suppl. 1 to Acquisition and Support Procedures for Computer Resources
AFR 800-14, V II in Systems. 18 October 1976.
- AFSCM/AFLOM 375-7 Configuration Management for Systems, Equipment,
Munitions, and Computer Programs. 31 March 1971.
- AFSCP 800-3 A Guide for Program Management. 9 April 1976.
- MIL-STD-480 Configuration Control - Engineering Changes, Deviations,
and Waivers. 30 October 1968.
- MIL-STD-483 (USAF) Configuration Management Practices for Systems,
Equipment, Munitions, and Computer Programs. 12 April 1971.
- MIL-STD-490 Specification Practices. 30 October 1968.
- MIL-STD-499A (USAF) Engineering Management. 1 May 1974.
- MIL-STD-881A Work Breakdown Structures for Defense Materiel Items.
25 April 1975.
- MIL-STD-1521A (USAF) Technical Reviews and Audits for Systems, Equip-
ments, and Computer Programs. 1 June 1976.
- MIL-S-52779 (AD) Software Quality Assurance Program Requirements.
5 April 1974.

8.2 TECHNICAL REPORTS

Bolen, N. E. An Air Force Guide to Contracting for Software Acquisition. ESD-TR-75-365; Electronic Systems Division, AFSC, Hanscom AFB, Mass. January 1976. (NTIS Accession No. AD-A020444)

Connolly, J. T. Software Acquisition Management Guidebook: Regulations Specifications and Standards. ESD-TR-75-91; Electronic Systems Division, AFSC, Hanscom AFB, Mass. October 1975. (NTIS Accession No. AD-A016401)

Glore, J. B. and Bjerstedt, W. R. Software Acquisition Management Guidebook: Statement of Work Preparation. ESD-TR-77-16; Electronic Systems Division, AFSC, Hanscom AFB, Mass. January 1977. (NTIS Accession No. AD-A035924)

Glore, J. B. Software Acquisition Management Guidebook: Life Cycle Events. ESD-TR-77-22; Electronic Systems Division, AFSC, Hanscom AFB, Mass. February 1977. (NTIS Accession No. AD-A037115)

Hagan, S. R. and Knight, C. W. An Air Force Guide for Monitoring and Reporting Software Development Status. ESD-TR-75-85; Electronic Systems Division, AFSC, Hanscom AFB, Mass. September 1975. (NTIS Accession No. AD-A-016488)

Peterson, D. R. Software Acquisition Management Guidebook: Software Development and Maintenance Facilities. ESD-TR-77-130; Electronic Systems Division, AFSC, Hanscom AFB, Mass. April 1977. (NTIS Accession No. AD-A038234)

Schoeffel, W. L. An Air Force Guide to Software Documentation Requirements. ESD-TR-76-159; Electronic Systems Division, AFSC, Hanscom AFB, Mass. June 1976. (NTIS Accession No. AD-A027051)

Searle, L. V. and Henderson, R. L. System Engineering Guide for Computer Programs. ESD-TR-68-1; Electronic Systems Division, AFSC, Hanscom AFB, Mass. March 1968. (DDC Accession No. AD669-129)

Searle, L. V., Rosove, P. E., and Sydow, E. H. Systems Management Applied to Large Computer Programs in BUIC III; Review of Experience. ESD-TR-69-302; Electronic Systems Division, AFSC, Hanscom AFB, Mass. June 1969. (DDC Accession No. AD699-585)

Searle, L. V. and Neil, G. Configuration Management of Computer Programs by the Air Force: Principles and Documentation. AFIPS Conference Proceedings, Spring Joint Computer Conference, Vol. 30, 1967, pp. 45-49.

SECTION 9. TERMS AND ABBREVIATIONS

9.1 TERMS*

Adaptation Data. Data whose values are fixed for a given site or mission, but which may vary for different sites or missions. Adaptation data represent one class of data that may be contained in the data base of a computer program configuration item designed for multiple-site or multiple-mission uses.

Addendum Specification. (See Configuration Item Specification Addendum.)

Advance Change Notice (ACN). The ACN is a document (e.g., AFSC Form 223) which precedes the preliminary or formal ECP, contains information establishing the need for the change, and allows for effective initial evaluation of the suggested change. (See AFSCP 375-1.) (MIL-STD-483)

MIL-STD-483 provides instructions for applicability of the ACN (or ACSN) to computer programs (para. 140.3.1) which duplicate those provided for equipment CIs (para. 130.3). However, the form is rarely if ever used for CPCIs. It does not become applicable, according to the instructions, until after the product baseline is established; it is not applicable even then unless specifically required in the given contract; and, in practice, it is only one of various optional ways in which study and coordination can be accomplished as a basis for procuring activity authorization to prepare a formal ECP.

Advance Change/Study Notice (ACSN). See Advance Change Notice (ACN).

Allocated Baseline. (See Baseline.)

Allocated Configuration Identification (ACI). Current approved technical documentation defining performance, design, and qualification requirements for a configuration item. In effect, the development (Type B) specification or equivalent.

*The DoD configuration management standard referenced in AFR 65-3 as "MIL-STD-CMX" is in process of Joint Services review and coordination at the time of this writing as MIL-STD-XXX. A coordination draft of MIL-STD-XXX contains a comprehensive glossary of terms to be standardized for DoD-wide uses, from which a number of these definitions were drawn. The draft is not a legitimate reference. It is identified herein, however, for the purpose of acknowledging the actual source of those particular definitions.

Baseline. The configuration of a configuration item (CI) as defined by an identification document or set of such documents formally designated and fixed at a specific time during a CI's life cycle. Baselines, plus approved changes from those baselines, constitute the current configuration identification. For configuration management there are three baselines, as follows:

Functional Baseline. The initial approved functional configuration identification.

Allocated Baseline. The initial approved allocated configuration identification.

Product Baseline. The initial approved or conditionally approved product configuration identification. (AFR 65-3)

Basic Issue. The first issue of a newly-developed document that is submitted for acceptance in finished form, often preceded by one or more draft issues. In the case of a specification, the basic issue is the first formal issue that bears authenticating signatures on the title page and defines the initial allocated or product baseline for the given item.

Change Issue Identifier. An alphanumeric number assigned by a developer to identify successive updates to specifications or other documents by means of change pages, as distinct from complete revisions. For a specification, the change issue identifier can be equivalent to the SCI number.

Computer Program. An ordered set of instructions and data required to control the operation of a computer. The end product of the process required to produce a computer program is usually a punched deck of cards, magnetic or paper tapes, or other physical media containing the ordered set of instructions in a form suitable for insertion into a computer. Under control of the instructions, the computer performs a set of well-defined and logically related functions. (MIL-STD-XXX)

Computer Program Component (CPC). A functionally or logically distinct part of a computer program distinguished for purposes of convenience in designing and specifying a complex computer program as an assembly of subordinate elements. (MIL-STD-483)

Computer Program Configuration Item (CPCI). A computer programming end product whose development and subsequent modification are subject to configuration management. (MIL-STD-XXX)

Computer Program Development Specification. A document which specifies the total functional performance requirements for each CPCI. This specification represents a comprehensive and definitive statement of the performance, design, and test requirements to be met by the computer program (MIL-STD-XXX)
Equivalent to "Part I CPCI specification" or "Type B5 specification".

Computer Program Product Specification. A document or series of documents which contain the detailed technical description of the CPCI as designed and coded. It is a complete description of all routines, limits, timing, flow, and data base characteristics of the computer program, including listings of the coded instructions. (MIL-STD-XXX)
Equivalent to "Part II CPCI specification" or "Type C5 specification".

Configuration. The functional and/or physical characteristics of hardware/computer programs as set forth in technical documentation and achieved in a product. (AFR 65-3)

Configuration Control. The systematic evaluation, coordination, approval or disapproval, and implementation of all approved changes in the configuration item after formal establishment of its configuration identification. (AFR 65-3)

Configuration Control Board (CCB). A board composed of representatives of various functional organizations used to (1) serve as a body to review, verify classification of, and approve/disapprove proposed changes and deviations, and (2) to perform total impact evaluation of proposed engineering changes.
(MIL-STD-XXX)

Configuration Control Board Directive (CCBD). A document which records the decision of the configuration control board (CCB) approval or disapproval of a proposed change submitted by a contractor, and is the basis for issuance of a contract modification if the change is approved. (MIL-STD-XXX)

Configuration Identification. The current approved or conditionally approved technical documentation for a configuration item as set forth in specifications, drawings, and associated lists, and documents referenced therein.
(MIL-STD-XXX)

Configuration Item (CI). An aggregation of hardware/computer programs or any of its discrete portions, which satisfies an end-use function and is designated by the Government for configuration management. CIs may vary widely in complexity, size, and type, from an aircraft, electronic, or ship system to a test meter or round of ammunition. (Abbreviated, from AFR 65-3)

Configuration Item Addendum Specification. A document prepared by writing a specification (addendum) by direct reference to an existing specification and recording in the new specification, reference to each paragraph in the existing specification. A specification created in this manner is a new and complete specification with a new specification number. (MIL-STD-XXX)

Configuration Item Development Record. A record which provides status information on the development progress of a configuration item. (MIL-STD-483)
For computer programs, equivalent to Section A of the configuration index.

Configuration Management. A discipline applying technical and administrative direction and surveillance to (1) identify and document the functional and physical characteristics of a configuration item (2) control changes to those characteristics, and (3) record and report change processing and implementation status. (AFR 65-3)

Contract. The legal agreement between DoD and industry, or similar internal agreement wholly within the Government, for the development, production, maintenance, or modification of an item(s). (MIL-STD-XXX)

Critical Design Review (CDR). [MIL-STD-XXX provides separate definitions of CDR for hardware and computer programs, as follows:]

Critical Design Review (Hardware). A formal technical review of the design of an item to assure that design requirements have been met before release of documentation for production.

Critical Design Review (Computer Program). A formal technical review of the design as depicted by the specification and flow diagrams, sufficiently detailed to enable the programmer to code, compile, and debug a computer program, to assure that design requirements have been met before beginning coding.

Critical Item. An item within a configuration item (CI) which, because of special engineering or logistic considerations, requires an approved specification to establish technical or inventory control at a level below the CI level. (MIL-STD-XXX)

The critical item designation does not apply to computer programs.

Data Item Description (DID). A standard form (DD Form 1664) employed to define format and content requirements for specifications, reports, manuals, and various other items of technical or management data to be delivered under a contract.

Development Test and Evaluation (DT&E). Test and evaluation conducted by the procuring command and its contractors to demonstrate that the design and development process is complete and that the system will meet specifications.

Deviation. A specific written authorization, granted prior to the manufacture of an item, to depart from a particular performance or design requirement of a contract, specification, drawing, or other document for a specific number of units or a specific period of time. A deviation differs from an engineering change in that an approved engineering change requires corresponding revision of the documentation defining the affected item, whereas a deviation does not contemplate revision of the applicable specification or drawing. (MIL-STD-480)

As they pertain to equipment CIs, deviations and waivers are documented discrepancies between the actual configuration of one or more units of the CI and the configuration defined in the CI's specification. The principal difference is that the deviation is processed in advance of manufacture, whereas the waiver is granted during production or at the time of inspection by the procuring activity. The specification referenced is typically the product specification, together with its associated engineering drawings and design/construction standards.

Hence, deviations and waivers used for those purposes do not have any comparable applicability to computer programs, basically because the CPCI product specification does not govern the "manufacture" of units in a comparable manner. Using the term in its general sense, deviations may often occur for units (copies) of a CPCI, as described in 5.4.1 and 6.2 of the text herein. Those should not occur except when authorized by proper authority. However, such deviations are not normally associated with procuring activity acceptance of the CPCI, or versions thereof, either before or at the time of its delivery. If a given deviation is authorized (e.g., for test purposes) and proves desirable to retain, the normal solution is to process an engineering change to the specification.

Engineering Change. An alteration in the configuration of a configuration item or items, delivered, to be delivered, or under development, after formal establishment or its configuration identification, which results in a corresponding change in its descriptive documentation. (MIL-STD-480)

Engineering Change Proposal (ECP). A term which includes both a proposed engineering change and the documentation by which the change is described and suggested. (MIL-STD-480)

Functional Baseline. (See Baseline)

Functional Configuration Audit (FCA). A formal audit to validate that the development of a configuration item (CI) has been completed satisfactorily and that the CI has achieved the performance and functional characteristics specified in the functional or allocated configuration identification. (MIL-STD-XXX)

Functional Configuration Identification (FCI). The current approved or conditionally approved technical documentation which describes performance, design, and test requirements for a system and, if any, its system segments. In effect, the system specification or equivalent.

Generation Breakdown. A listing of subordinate items and parts comprising a system or major configuration item. The subordinate elements are listed in top-down order, reflecting their indented relationships in the assembly hierarchy as a whole.

Interface. A region common to two or more elements, systems, projects, or programs, characterized by mutual physical, functional, and/or procedural properties. (MIL-STD-XXX)

Interface Control. Interface control comprises the delineation of the procedures and documentation, both administrative and technical, contractually necessary for identification of functional and physical characteristics between two or more CIs which are provided by different contractors/Government agencies, and the resolution of problems thereto. (MIL-STD-483)

Interface Control Document (ICD). A document which records the compatible design or operating relations between two or more interfacing configuration item designs, and when approved, reflects the agreement between two or more contractors/Government agencies/contractor divisions. The documents are used as design control documents, delineating interface engineering data coordinated for the purpose of:

- (a) establishing and maintaining compatibility between co-functioning items;
 - (b) controlling interface design thereby preventing changes to items requirements which would affect compatibility with co-functioning subsystems;
 - (c) communicating design decisions and changes thereto to participating activities.
- (MIL-STD-100A)

Item. A nonspecific term used to denote any product, including systems, materials, parts, subassemblies, sets, accessories, etc. (MIL-STD-280A)

Notice of Revision (NOR). A form used to propose revisions to a drawing or list, and, after approval, to notify users that the drawing or list has been, or will be, revised accordingly. (Abbreviated from MIL-STD-480)

Applicability of the NOR to computer programs is provided for in paragraph 140.14 of MIL-STD-483. The NOR is comparable to the SCN, in that it may accompany an ECP, replacing the SCN, to describe the change to a specification (including, for equipment, engineering drawings) when it is desired to alter the configuration of certain items. MIL-STD-480 does not clarify how or whether it applies to Class II as well as to Class I changes; however, if used for a computer program change, it would seem to be appropriate for both. The NOR is prepared instead of an SCN when the contractor preparing the ECP is not the originator or custodian of the specification--as he would not be for an item purchased from a commercial vendor or perhaps for an inventory item. According to MIL-STD-480, the procuring activity then arranges (if he approves the NOR) to have the originator/custodian issue the change to the specification. --Thus, applicability to a computer program depends to some extent on relationships of both procuring activity and contractor to the item originator/custodian, and may involve legal considerations pertaining to data rights.

Operational Test and Evaluation (OT&E). Test and evaluation conducted by the using command and AFTEC to estimate a system's military utility, operational effectiveness, and operational suitability.

Physical Configuration Audit (PCA). The formal examination of the "as built" configuration of a unit of a CI against its technical documentation in order to establish the CI's initial product configuration identification. (MIL-STD-480)

Preliminary Design Review (PDR). A formal review of the preliminary design of a system functional area or of a configuration item to establish system compatibility of the design, identify specific engineering documentation and define physical and functional interface relationships. (MIL-STD-480)

Product Baseline. (See Baseline.)

Product Configuration Identification (PCI). In effect, the product specification for a CI, or its equivalent documentation.

For equipment CIs: The current approved or conditionally approved technical documentation which defines the configuration of a CI during the production, operation, maintenance, and logistic support phases of its life cycle, and which prescribes (1) all necessary physical or form, fit, and function characteristics of a CI, and (2) the selected functional characteristics designated for production acceptance testing, and (3) the production acceptance tests. (MIL-STD-480)

For computer program CIs: Technical computer programming documentation which describes the "as built" design and coding of the computer program.

Qualification. Verification by means of tests and other suitable methods that a newly-developed item meets the requirements of its development (Type B) specification.

Retrofit. Incorporation of an engineering change (at any level) in accepted or in-service items. (MIL-STD-480) As related to the transfer of program management responsibility from AFSC to AFLC, AFR 57-4 distinguishes two major classes of retrofit changes as follows:

Modifications. Changes for which the requirements are identified after PMRT.

Updating Changes. Changes for which requirements are identified prior to PMRT, but which may not be implemented until after PMRT; AFSC is normally responsible for implementing this class of changes.

Revision.

- (a) A new issue of an entire document or volume which completely supersedes any previous issue, all pages being identified by the same applicable revision element of the document number.
- (b) Generally, a change to a document or volume made by any suitable method.

Serialization. The application of numeric and/or alphabetic designators in a specified order to distinguish individual units of a CI. (MIL-STD-XXX)

Software. In this guidebook, synonymous with "computer program(s)". (See 1.6 and 7.2 herein.)

Specification. A document which clearly and accurately describes the essential technical requirements for items, materials, or services including the procedures by which it will be determined that the requirements have been met. (DoDD 4120.3)

Specification Change Notice (SCN). A document used to propose, transmit and record changes to a specification. In proposed form, prior to approval, the SCN (P) supplies proposed changes in the text of each page affected. (MIL-STD-480)

Specification Tree. A diagram or listing showing the indentured relationships between specification-type documents or requirements documents independent of the assembly or installation relationships of the items affected.

(MIL-STD-XXX)

System A composite of equipment, skills, and techniques capable of performing or supporting an operational role, or both. A complete system includes all equipment, related facilities, material, computer programs, services, and personnel required for its operation and support to the degree that it can be considered a self-sufficient unit in its intended operational environment.

(MIL-STD-XXX)

System Allocation Document. A document which identifies the segregation of configuration items by serial number and the system configuration at each location. (MIL-STD-483)

The SAD has been applied, at the system and system segment levels, with the requirement to include computer programs in accordance with Appendix XI of MIL-STD-483. Its real usefulness is evidently limited, however, to providing a record of the fact that given CPCIs are assigned to the identified location. Emphasis in the form is on numbers and specific identification of equipment units (serial numbers) and on top drawing and part numbers. Much of that information required for equipment is either not applicable or irrelevant to computer programs--especially if the given location has the capability, as many do, to duplicate its own additional copies of the CPCIs as needed.

System Segment. A discrete package of system performance requirements, functional interfaces, and configuration items contracted to one contractor or assigned to one Government organization directly responsible to the procuring agency for that part of the system's total performance. (MIL-STD-483)

Unit. In configuration management, one complete article of a configuration item. For example, one F111A of a total quantity of 100 F111As. (MIL-STD-480) By analogy, individual copies of a master tape would constitute units of a CPCI. Units of an equipment CI are distinguished by their serial numbers, which are important in configuration status accounting. The established hardware practices and concepts pertaining to units and serialization tend not to have comparable significance or uses in software configuration management.

Version. The actual configuration of a computer program configuration item (CPCI) which is introduced for installation and test or operation into the system in the form of a magnetic tape, deck of cards, disc, or other physical medium. A new version is created (a) when a newly-developed item is first delivered; or (b) whenever the CPCI is completely reassembled, containing all Class I and Class II changes accomplished since the preceding version.

An interim version occurs when a single class I change is introduced into an existing version through delivery of partial changes to the CPI, short of complete reassembly or release of a complete raw tape or card deck.

waiver. A written authorization to accept a configuration item or other designated items, which during production or after having been submitted for inspection, are found to depart from specified requirements, but nevertheless are considered suitable for use "as is" or after rework by an approved method. (Also, see Deviation.) (MIL-STD-480)

9.2 ABBREVIATIONS

ACI	Allocated Configuration Identification
AFLC	Air Force Logistics Command
AFR	Air Force Regulation
AFSC	Air Force Systems Command
AFTLC	Air Force Test and Evaluation Center
ASD	Aeronautical Systems Division (AFSC)
ATC	Air Training Command
CCB	Configuration Control Board
CCBL	Configuration Control Board Directive
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CI	Configuration Item
CMO	Configuration Management Office
CPC	Computer Program Component
CPCI	Computer Program Configuration Item
CPCSB	Computer Program Configuration Sub-Board
CPDP	Computer Program Development Plan
CPTN	Computer Program Identification Number
CPT&E	Computer Program Test and Evaluation
CR	Change Report (Class II)
CRISP	Computer Resources Integrated Support Plan
DCN	Document Change Notice
DID	Data Item Description
DoD	Department of Defense
DT&E	Development Test and Evaluation
ECP	Engineering Change Proposal
ESD	Electronic Systems Division (AFSC)

FCA	Functional Configuration Audit
FCI	Functional Configuration Identification
FQR	Formal Qualification Review
ICD	Interface Control Drawing
ICWG	Interface Control Working Group
LEP	List of Effective Pages
NOR	Notice of Revision
O/S CMP	Operational/Support Configuration Management Procedures
OT&E	Operational Test and Evaluation
PCA	Physical Configuration Audit
PCI	Product Configuration Identification
PDR	Preliminary Design Review
PMD	Program Management Directive
PMRT	Program Management Responsibility Transfer
PO	Program Office
RFP	Request for Proposal
ROC	Required Operational Capability
SAMSO	Space and Missile Systems Organization (AFSC)
SCN	Specification Change Notice
SDR	System Design Review
SOW	Statement of Work
SRR	System Requirements Review
TCD	To Be Determined
TO	Technical Order
TR	Technical Report
VDD	Version Description Document