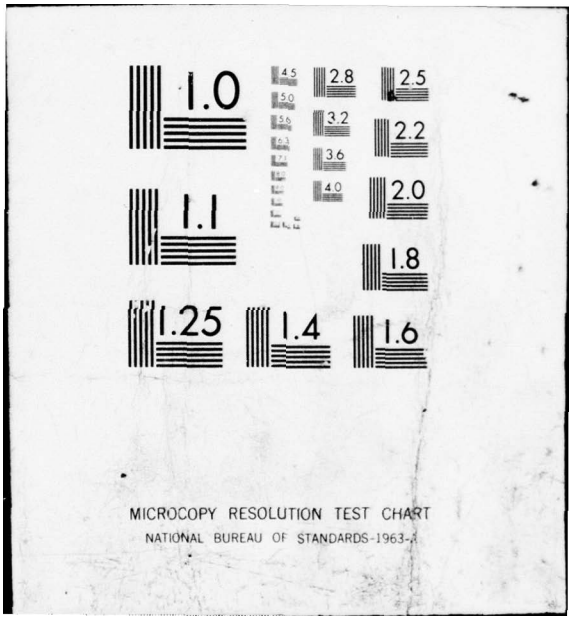
AD-A052 126 DEFENSE SYSTEMS MANAGEMENT COLL FORT BELVOIR VA F/G 9/2
APPLYING PROGRAM MANAGEMENT CONCEPTS TO SOFTWARE DEVELOPMENT: A--ETC(U)
NOV 77 R CAUDILL

UNCLASSIFIED NL

1 OF 2
AD A052 126


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DEFENSE SYSTEMS MANAGEMENT COLLEGE

PROGRAM MANAGEMENT COURSE
INDIVIDUAL STUDY PROGRAM

APPLYING PROGRAM MANAGEMENT CONCEPTS
TO SOFTWARE DEVELOPMENT: AN
AFR 300-15 CRITIQUE

STUDY PROJECT REPORT
PMC 77-2

Ray Caudill

GS-13 USAF

FORT BELVOIR, VIRGINIA 22060

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) APPLYING PROGRAM MANAGEMENT CONCEPTS TO SOFTWARE DEVELOPMENT: AN AFR 300-15 CRITIQUE		5. TYPE OF REPORT & PERIOD COVERED Study Project Report 77-2
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) RAY CAUDILL		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS DEFENSE SYSTEMS MANAGEMENT COLLEGE FT. BELVOIR, VA 22060		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS DEFENSE SYSTEMS MANAGEMENT COLLEGE FT. BELVOIR, VA 22060		12. REPORT DATE 1977-2
		13. NUMBER OF PAGES 96
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

UNLIMITED

DISTRIBUTION STATEMENT A
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)

SEE ATTACHED SHEET

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

SEE ATTACHED SHEET

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DEFENSE SYSTEMS MANAGEMENT COLLEGE
ABSTRACT

STUDY TITLE: APPLYING PROGRAM MANAGEMENT CONCEPTS TO SOFTWARE DEVELOPMENT: AN AFR 300-15 CRITIQUE

STUDY PROJECT GOALS:

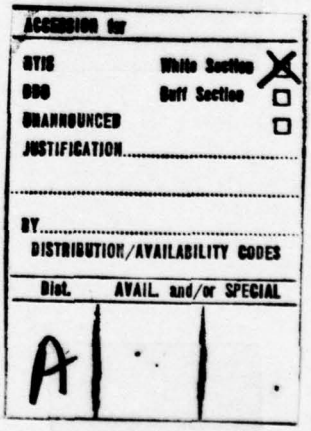
To identify and propose changes to AFR 300-15 in an effort to further adapt concepts and principles of program management to computer software development.

STUDY REPORT ABSTRACT:

The USAF has recently incorporated concepts and principles of computer software with preparation of AFR 300-15. However, it is believed that those concepts were not tailored finitely enough. Further, AFR 300-15 appears to adapt the development process to fit the principles rather than the reverse.

This paper proposes changes to AFR 300-15 which tailor/adapt the principles and concepts of program management to the software developmental process. This is accomplished by realignment of some functions within the phases, by renaming baselines and reviews, by shifting reviews, and by establishing some new baselines and reviews.

SUBJECT DESCRIPTORS:
Computer Resources Management (10.02.09)



NAME, RANK, SERVICE	CLASS	DATE
Ray Caudill - USAF - GS-13	77-2	25 Oct 77

APPLYING PROGRAM MANAGEMENT CONCEPTS TO SOFTWARE DEVELOPMENT: AN AFR 300-15 CRITIQUE

Individual Study Program
Study Project Report

Defense Systems Management College
Program Management Course
Class 77-2

by

Ray Caudill
GS-13 DAFC

November 1977

Study Project Advisor
Lt/C Joe Arcieri, USAF

EXECUTIVE SUMMARY

1. Historically, the computer software development process has been difficult for managers to control. Systems design and programming efforts were frequently thought of as highly individualistic and artistic endeavors. These phenomonea are doubtlessly attributable tc the relatively short, yet explosive, life of the Electronic Data Processing industry.

2. With personnel costs rising drastically, and with hardware costs rising at a lesser pace, ways must be found to reduce software costs as well as reduce the attendant risks.

3. Industry and trade literature is replete with "cure-alls". However, the U.S. Air Force has chosen to approach the problem by applying the program management principles and concepts of phasing, baselining, and formalized reviews and audits to computer software development. This has the property of causing the developmental cycle to be broken into its various component parts and each attacked separately, yet with integration in mind. This approach is documented in AFR 300-15 Automated Data System Project Management. It is an adaptation of the practices widely employed by DOD and industry in weapons systems acquisition.

4. This paper is a critique of AFR 300-15. This critique results in specific proposed changes to AFR 300-15. This paper will be presented to Hq. USAF/KRA upon the author's graduation from DSMC. The changes include the following:

a. Changes in "purposes" paragraphs to change the "philosophy" of the document.

b. Redefinition of some phases.

c. Addition of some new formal reviews.

d. Realign or refocus existing reviews.

In summary, these changes will cause the document to reflect the methodology shown in figure 1.

Conceptual	Definition	Development		Test	Operation
		Design Segment	Program Segment		
Prepare and approve the concept.	Fully define the problem or situation from user viewpoint.	Design the software to program level with all inter-faces	Program	Test	Operate
Concept Certification Review	Functional Definition Review	System Design Review	Program Logic Review	System Review	
Concept Baseline	Functional Baseline	Design Baseline	Product Verification Review		
				Unchanged	Unchanged

Figure 1

TABLE OF CONTENTS

Section I

Background

1. "Today, providing computer software involves greater cost and risk than providing computer equipment, because hardware is mass produced by industry using proven technology, while software is still produced mostly by the craft of individual computer programmers and users. Over 70% of all software is developed by the users of computers, rather than the manufacturers of computer equipment or the producers of commercial software. Software is very expensive and the development costs for new systems are growing to enormous levels. The Federal government is estimated to spend as much as $5 billion annually on software, and its accumulated investment in current software probably exceeds $25 billion. Software costs are projected to become 9 times as great as hardware costs by 1985" (12).

This quote is only one among many which could be cited which attest to the magnatude of the cost, schedule and technical "risks" associated with computer software. Management at all levels appears to be taking a hard look at the software developmental process in an effort to find ways to better control that process.

2. In his article "Why Projects Fail", Stephen Keider says "One of the primary causes for the failure of data processing projects is that such projects are often not initially defined...Once a project has begun, no one seems to know:

- how the project was started;
- what the staffing is, or was, at any point in time;
- what activities have been performed;
- when the project will end;
- what the project will accomplish.

Essentially, because projects are rarely formally defined, they are rarely completed." (4:275).

3. There is still another factor which contributes to problems associated with computer software development. That problem is the same one that might be applied to almost any endeavor--management visibility and control. Perhaps if management had control of and visibility into the development process, the problems in the above paragraphs would be minimized. Major Cecil Martins visited many Automatic Data Processing installations in the government, private industry and universities compiling data for a PHD dissertation. In a presentation summarizing his findings, he stated that poor management visibility and control are to blame for most of the problems which pervade the system development process. Further, this research indicated that those organizations which have adopted an orderly, step-by-step approach to software development exhibit fewer problems than do those who have not "disciplined" the effort (13).

4. The thesis of this paper is based upon the aforementioned problems. Namely, that management visibility and control are somewhat lacking and that that lack allows, even encourages, other problems. It is further argued that conventional manage-

ment practices can cope with the software development process once the process is recognized as being composed of several component parts, each with a purpose, which together constitute the total computer software developmental process. Recognizing the component parts allows the manager and the technician to transcend the whole, thereby gaining an insight which allows understanding, does away with "artistry", and allows both management and technical discipline.

5. A brief discussion of how and why the software developmental process arrived at its present state is warranted at this point. First, the computer industry is relatively infantile in terms of years. However, its growth has been phenomenal. People in the trade often obtained new hardware even before they had become accustomed to the old. This gave rise to a rather common opinion that conventional management practices cannot be applied to software development. This opinion which still persists has its roots in the very early days of computers. In the fifties and early sixties, computer programmers and systems analysts were generally left to accomplish their work in any style and manner each chose. They were generally exempt from management control. In short, design and programming were generally free style. Many Automatic Data Processing personnel relished this opportunity to be artistic and individualistic made possible by the near abdication of traditional management responsibilities. In a 1973 Datamation article, Boehme quotes an "Air Force Decisionmaker" as that decisionmaker refers to ADP people as follows:

> .....You software guys are too much like the weavers in the story about the Emperor and his new clothes. When I go out to check on a software development the answers I get sound like, 'We're fantastically busy weaving this magic cloth. Just wait awhile and it'll look terrific.' But there's nothing I can see or touch, no numbers I can relate to, no way to pick up signals that things aren't really all that great. And there are too many people I know who have come out at the end wearing a bunch of expensive rags or nothing at all (1:48).

6. Still another author, Richard L. Nolan writes the following:

> Top management has never been totally comfortable with the (EDP) Electronic Data Processing manager. And it is fair to say that the EDP manager has not always completely understood the rationale for top management's actions in regard to EDP." Nolan continues, "Not so subtle is the continuing high attrition rate among those who hold top EDP spots. I have estimated that annual EDP management turnover in 1972 was from 40% to 50%. In a more recent 12 month period, 25% to 35% of the EDP managers in more than 450 companies were replaced, according to a survey conducted by a major vendor of computed hardware and software in 1975. These rates compared to 10% to 15% turnover rates among other senior managers (8:402).

7. Perhaps the most notable result of this brief history of disorganization, the one which causes most of today's developmental problems, is that no standard developmental methodology, based upon a series of stages or phases of developmental effort, is in widespread use. The Nolan article mentioned above "talks" at great lengths about management but never really describes, or mentions what it is that's being managed. Without a standard developmental methodology, there is little hope of gaining the much needed management visibility and control over the software developmental process.

Section II

Present Situation

1. The Air Force has recognized the risks involved in computer software development and has taken steps toward resolution with its AFR 300-15 Automated Data System Project Management. That document transcends the "whole" and does identify the component parts of the software developmental process, trying to make them visible and controllable. Essentially, AFR 300-15 employs the principles and concepts of program management (phases, baselines, formal reviews) as practiced by the Department of Defense in the acquisition of weapons systems.

2. The application of program management principles and concepts to computer software development is a significant step forward but the tailoring of those principles may not have gone far enough. In short, there seem to be places in AFR 300-15 where the logical developmental process was shaped to fit program management rather than the reverse.

3. The purpose of this paper is to propose changes to AFR 300-15 to further refine/adapt the ideals of phasing, baselining, reviewing and auditing to the logical and natural computer software developmental process.

Section III

Recommended Changes to AFR 300-15

1. This section contains specific proposed changes to AFR 300-15. At the top of each change is a page number which corresponds to the page in AFR 300-15. AFR 300-15 is included as attachment 1 for reference purposes. Rationale for each change preceeds it, usually on the same page for shorter changes. In the case of massive changes, the entire paragraph was rewritten.

Rationale: This is a change to the title/purpose page of the regulation. This change will alter the intent of the entire document. Currently, this page states that this document provides "...a methodology for the management of data system (ADS) projects." This document is far more than a "methodology of management". This document prescribes a "software development methodology". Once that software developmental methodology is established, traditional management principles and practices become workable.

Change: Remove the first sentence and insert "This regulation outlines policies and procedures which provide a standard software developmental methodology. Management of that methodology is enhanced through a visible developmental life cycle, formal reviews, baselines, and change control procedures. It must be used ... (continue)"

Rationale: This chapter is essentially an introduction to the document. There is only one recommended change to this chapter. The rationale is the same as that stated for the change to page 1.

Change: Replace paragraph 1-1 with: "The purpose of this regulation is to prescribe a standardized software developmental methodology. The software developmental life cycle is fully described in terms of phases and associated deliverable products. Management of the software developmental life cycle is greatly enhanced by the requirement for formal reviews, baseline establishment, and other factors designed after the engineering discipline of configuration management."

PROPOSED CHAPTER 2

OF AFR 300-15

Note: Entire Chapter Rewritten

Chapter 2

Rationale: This is a key chapter in that it describes the software developmental process in detail. In short, this chapter establishes the entire software developmental cycle by identifying, for the first time, the component parts or anatomy of the developmental life cycle. This entire chapter was rewritten and with it the philosophy of the regulation changes from a mere management document to a software development/procedural document. The following changes were made:

a. Phase activities realigned. Several phase activities were realigned so that they now occur in an entirely different phase. Other activities were changed or replaced which forces more discreteness between phases. It is here that the process takes on some discipline and management achieves control and visibility.

b. Reviews were renamed, realigned and a new one was added. Some formal reviews had names which were short on meaningfulness in the data processing community. The names were changed to more accurately reflect their function. Secondly, some were realigned to occur at a different point in the cycle.

c. Baselines. Baselines were moved, renamed and a new one was added. Like the reviews, names were not meaningful. They were renamed to more clearly reflect their purpose. A baseline was added after the concept phase as none was there before. It is as important to baseline this phase as it is to baseline any other.

CHAPTER 2

SOFTWARE DEVELOPMENT METHODOLOGY

2-1. Purpose. The purpose of this paragraph is to describe the software development methodology. Essentially, that methodology consists of movement through a series of phases. Each phase has a purpose and fulfillment of that purpose is the basis for action in the next phase. Baselines serve the purpose of establishing the fact that specific actions have been accomplished, that change control procedures are to be used thereafter, and to serve as the point of departure for further effort. Baselines are established or denied at reviews.

2-2. CONCEPTUAL PHASE. The primary purposes of the actions in this phase are to fully document the requirement at conceptual level and to gain approval to proceed with further exploration of the problem or definition requirement. This approval is concept certification, which means that the approval authority (AFR 300-2 approval levels) concurs with the conceptualized solution to the stated problem or requirement.

a. TASKS. The first task is the preparation of the Data Automation Requirement as per the instruction in AFR 300-12. In order to prepare this document, feasibility studies, functional systems analysis, etc. may be undertaken. Care must be taken to guard against designing a software system at this point. Secondly this document should go into only the detail required to gain concept certification. Full definition comes in the next phase. The approval authority will issue the Data Project

Directive. The purpose of this document is to formalize concept certification and to direct that the project proceed through subsequent phase(s).

b. DOCUMENTS:

(1) Data Automation Requirement

(2) Data Project Directive

c. REVIEW: The Concept Certification Review will be performed at the conclusion of this phase. It may be performed by the AFR 300-2 approval authority or at a level prescribed by him. The results of the review and resultant recommendations will be forwarded to the AFR 300-2 approval authority for final approval and granting of concept certification. In this case, a letter of concept certification will be signed by the approval authority and the letter will be appended to the DAR. Any DAR without such a letter will be considered as not having been certified. Concept certification automatically establishes the Concept Baseline and change control procedures automatically are invoked relative to the DAR.

2-3. DEFINITION PHASE. The purpose of the Definition Phase is to fully define the functional aspects of the system to be developed. This is not to be construed as a software system design, which is the purpose of the next phase. It is the purpose of this phase to take the DAR prepared in the previous phase as the basis and prepare a Functional Description which requires the most detailed functional definition of the requirement. It is logical that the software system be conceptualized during this

phase and the Functional Description provides for that.

a. TASKS

(1) Prepare the Data Project Plan

(2) Prepare the Functional Description

(3) Begin drafting user, operation and maintenance manuals

b. Documents: (as specified in a. above)

c. Review. The Functional System Definition Review is performed at the close of this phase. The purpose of this review is to assure that the Functional Description portrays the functional specifications to the satisfaction of the user. Secondly, the ADP systems analysts must agree that the information provided is sufficient to serve as the basis for system design. Approval of the Functional Description automatically establishes the Functional Baseline and invokes change control procedures against the Functional Description.

2-4. DEVELOPMENT PHASE. The purpose of this phase is to design the software system and to design, code and test the computer programs. Accordingly, this phase is divided into two segments: design and program. The Design Baseline separates the two segments.

a. Design Segment Tasks and/or Documents:

(1) Prepare the system/subsystem specifications (constitutes the design of the software system)

(2) Prepare Data Requirements Document

(3) Prepare Data Base Specifications

b. Review: The System Design Review is performed to determine the adequacy of the system design. The capabilities of the design must accomplish the requirements established in the Functional Description. Upon favorable completion of this review, the Design Baseline is automatically established. The system/subsystem specifications are then covered by change control procedures.

c. Program Segment Tasks and/or Documents

(1) Prepare Program specifications

(2) Perform Program Logic Review iteratively until each program is approved for coding

d. Reviews.

(1) A Program Logic Review is performed on each Program Specification prior to the commencement of coding. The purpose is tc assure the adequacy of the program logic and to assure that the program will perform as specified by the system/subsystem specification.

(2) The Product Verification Review is the formal culmination of the physical and functional configuration audits performed throughout this phase. Completion of this review establishes the Product Baseline.

2-5. TEST PHASE. The installation and validation test of the system in the operational environment by the designated user representative or by an independent agency.

a. VALIDATION TESTING. Conduct two-stage validation testing of the system by an independent test group. In the first

stage, perform in-house validation testing of the system requirements, correct problems, and update documentation. In the second stage, rerun validation test for the user, or review successful in-house validation test runs with the user. In addition, validate program specifications and other relevant documentation in both stages (paragraph 6-5).

b. SYSTEM VALIDATION REVIEW (SVR). The SVR validates that performance of a CPCI, as determined through validation testing, complies with the SS and the FD (paragraph 4-9).

2-6. OPERATION PHASE. The operation, maintenance, and product improvement by the user and/or the developer.

Rationale. Configuration management is a relatively new term to many people in the ADP community. Therefore, a definition of configuration management is in order in the lead paragraph on that subject. This change does that.

Change. Remove paragraph 3-1 and replace it with the following paragraph. "Configuration management is a term long associated with the development of weapons systems. The term now applies to the development of general purpose computer software as the concepts and principles of configuration management have been adapted to that usage. This document seeks to ensure that the software developmental life cycle is orderly, visible to management and users, is controllable and the outcome more predictable. To those ends, the software configuration management concepts of phases, baselines, formal reviews, and change control have been adapted and included herein. While the concepts and the principles are the same, the phases, baselines, reviews, and change control concepts have been revised, shifted, tailored and otherwise changed to make them "fit" the logical software developmental life cycle. Within the AFR 300-series software developmental area configuration management is the discipline applied to the developmental life cycle which gives it a visible, comprehensible structure (phase), enhances control (baselines, reviews) and which enhances developmental changes (change control).

Chapter 4

Rationale: The changes in Chapter 2 incorporated some new reviews. This chapter includes the details for accomplishing reviews. There is no additional rationale beyond what was provided for Chapter 2. Review additions or changes are:

4-3. Concept Certification Review - New

4-4. Functional System Definition Review - Name change and partial change in function.

4-6. Program Logic Review - Name change

PROPOSED CHAPTER 4 of AFR 300-15

Note: Entire Chapter Rewritten

CHAPTER 4

Reviews and Audits

4-1. PURPOSE. The purpose of this chapter is to describe the reviews and audits that must be performed throughout the software developmental life cycle.

4-2. PARTICIPATION AND RESPONSIBILITIES. The project manager or his designee chairs all reviews and audits. The user representative will make sure that the functional requirements are being adequately met.

a. Project Manager Responsibilities.

(1) Establish the time, the place, and the agenda for the reviews and audits, and coordinate these arragements with all participants. Depending on the type of review or audit, schedules for incremental reviews may have to be prepared and coordinated with all participants.

(2) Prepare for each review or audit in sufficient depth consistent with project scope and size. Material to be reviewed must be submitted to participants prior to each review, allowing time for comment. Documents should be reviewed for technical rather than editorial content. Review participants should prepare design problem reports (DPRs) and submit them to the originator of the material being reviewed. The originators of the material will evaluate the DPRs and reply to them at the formal review. Another approach is to correct minor discrepancies prior to the review, advise participants of the disposition, and hold the major DPRs for discussion at the meeting.

b. Resolution of disagreements between the development activity and the user will be escalated to the next level of management for resolution.

4-3. CONCEPT CERTIFICATION REVIEW (CCR). The CCR is conducted during the conceptual phase to formally approve the concept by review and approval of the DAR.

a. Review Items. The principal items to be reviewed at the CCR are the Data Automation Requirement (DAR), and the Data Project Directive (DPD).

b. Postreview Action. After completion of the CCR, the minutes are distributed by the project manager. Notification of any required postreview action items, plus official acknowledgment of conduct and completion of the review, are provided to the user, the developer, and the appropriate ADP approval authority.

4-4. FUNCTIONAL SYSTEM DEFINITION REVIEW (FSDR). The FSDR establishes the functional baseline for individual CPCIs or for the entire system.

a. Review Items.

(1) The FSDR documentation package should include the following:

(a) Functional Description (FD) with changes

(b) Data Project Plan (DPP)

(2) The FSDR is conducted to accomplish:

(a) Review of the FD

(b) Insure the technical project risks are iden-

tified, ranked, avoided, and adequate tradeoffs are made.

(c) Insure a technical understanding of requirements has been reached and technical direction is provided to project participants.

(d) Insure project schedules and milestones are consistent with the allocated requirements.

b. Postreview Action. After completion of the FSDR, the minutes of the meeting are coordinated with all participants. Participants will also be advised of any postreview action items requiring response. Approval of the FD establishes the functional baseline.

4-5. SYSTEM DESIGN REVIEW (SDR). The SDR is a technical review of the system/subsystem design for a computer program configuration item (CPCI). Only one successful SDR per CPCI is required. A collective SDR for a functionally related group of CPCIs, treating each CPCI individually, may be held when advantageous to the project. The overall technical risks associated with each CPCI are also reviewed on a technical, cost, and schedule basis. NOTE: If the ADS is defined as a single CPCI, the SDR may be combined with the FSDR.

a. Review Items.

(1) The SDR documentation package should include the following:

(a) Updated functional description (FD)

(b) System/subsystem specification (SS)

(c) Draft of CPCI test plans (PT), less test

procedures

(d) Data base structure and organization

(e) Design problem reports (DPR)

(2) The SDR is conducted to establish the following:

(a) Compatibility of the design with the SS for the CPCI and with other CPCIs with which it must interface

(b) Adequacy of test plan

(3) To satisfy the intent of the SDR, the following tasks should be performed

(a) Review all functional interfaces. Review message formats, storage availability, and other considerations that may have been established in the SS. At this time, the interfaces between CPCI and hardware CIs should be defined at a level low enough to satisfy all software development needs.

(b) Review the structure of the CPCI as a whole, with emphasis on allocation of the functions delineated in the SS to individual computer programs; storage requirements and allocation; computer program operating sequences; and design of the data base

(c) Analyze critical timing requirements to insure the proposed CPCI design satisfies the timing requirements

(d) Review human factors impact of proposed sub-sustem design of the CPCI

(e) Review all changes to the SS subsequent to the established allocated baseline. Insure the FD and SS adequately reflect these changes

(f) Review the PT to insure it satisfies the requirements of the FD and SS

(g) Review status of all negative and provisional entries such as "not applicable" or "to be determined" in the SS

b. Postreview Action. After completion of the SDR, the minutes are published and distributed by the development activity. The user will be notified of any required postreview action.

4-6. PROGRAM LOGIC REVIEW (PLR).

a. Purpose. The PLR is a review conducted during the development phase before translating logic to coded instructions. The PLR insures the detailed design solution, as reflected in the draft program specification (PS), satisfies performance requirements established by the SS. The PLR also verifies system design compatibility with other CPCIs, and within the CPCI. Approval of the project manager is required to release portions of CPCIs for coding prior to PLR when necessary to maintain schedules.

b. Review Items and Tasks

(1) The PLR documentation should consist of the following:

(a) System/subsystem specification (SS), including approved changes

(b) Program specifications

(c) CPCI test plans (PT), including test procedures

(d) Design problem reports (DPR)

(e) System design review (SDR) minutes

(f) Draft of computer operation manual (OM)

(g) Draft of program maintenance manual (MM)

(h) Draft of the program specification (PS)

4-7. CONFIGURATION AUDITS

a. Purpose. A configuration audit is a process employed to verify conformance of a CPCI to specifications and standards. Two audits, the Functional Configuration Audit (FCA) and the Physical Configuration Audit (PCA), are required at two points during the ADS life cycle for an ADS requiring greater than fifty development manyears. The first point, prior to the Product Verification Review (PVR), is termed the preliminary FCA and PCA, and the second point is prior to the System Validation Review (SVR) and termed the final FCA and PCA. The preliminary audits are a thorough technical examination of the CPCI, while the final audits are directed at changes made to the CPCI as a result of formal testing. The following guidelines for conducting the FCA and PCA are provided.

b. FUNCTIONAL CONFIGURATION AUDIT (FCA). The FCA validates the CPCIs actual performance compliance with the SS. Test data is reviewed to validate that the CPCI performs as required. The project manager provides two FCA data packages to the organization performing the FCA, one to be delivered at some negotiated lead time prior to FCA, and the other to be made available during the FCA.

(1) Items to be provided before audit are:

(a) Nomenclature (name or descriptive title of the CPCI)

(b) Identification of necessary documentation

(2) The documentation package to be made available for FCA consists of:

(a) Program specification (PS), including listings

(b) System/subsystem specification (SS) with changes, if applicable

(c) Test plan (PT) and when available, test reports (RT)

(d) PLR and CDR minutes

(e) Status accounting records of SPRs, DBCRs, and BCRs

(3) The FCA consists of examining and evaluating:

(a) Test procedures and results to determine CPCI conformance with the SS and FD and quality assurance provisions

(b) Adequacy of analysis or simulations where performance parameters cannot completely be verified by test

(c) Any requirements stated in the FD that could not be met, and the developer's proposed solution

(d) All approved BCRs to insure they are incorporated and verified during testing

(4) Deficiencies noted and recommended corrective actions are documented in the FCA minutes.

c. PHYSICAL CONFIGURATION AUDIT (PCA). The PCA is a formal examination of the coded version of a CPCI against its technical documentation. The PCA includes a selective audit of the program specification (PS), including logic charts, listings, and design narrative. It also includes a review of the formal and completeness of any other documentation due for acceptance. The project manager provides two PCA data packages to the organization performing the PCA - one to be delivered at some negotiated lead time prior to PCA, and the other to be made available during PCA.

(1) Items to be delivered prior to audit consist of

(a) PCA date and location

(b) Identification of CPCIs to be accepted

(2) The documentation package to be provided and made available for PCA consists of:

(a) System/subsystem specification (SS)

(b) Program specification (PS)

(c) Test procedures (PT) and test report (RT)

(d) Draft computer operation manual (OM)

(e) Draft program maintenance manual (MM)

(f) CPCI tape and listings

(g) Applicable configuration management records

c. The PCA consists of reviewing:

(1) PS for format and completeness

(2) Selectively compare the listings with documentation

(3) Computer operation manual (OM) and program mainten-

ance manual (MM) for format and completeness

(4) The release and change control system to be sure it can properly control the processing and formal release of changes.

4-8. PRODUCT VERIFICATION REVIEW (PVR).

a. Purpose. PVR is conducted for each CPCI at the end of the development phase to establish the Product Baseline for that CPCI and to insure preparation for the test phase has been completed.

b. A preliminary functional configuration audit (FCA) and a preliminary physical configuration audit (PCA) will be conducted prior to the PVR to provide management with an independent technical assessment of the CPCI.

c. Items reviewed at the PVR include:

(1) Findings and recommendations resulting from the preliminary FCA and PCA

(2) Test plans

(3) Preparations and schedules for the test phase

d. Discrepancies which would lead to establishing an invalid product baseline and premature testing must be corrected prior to successful completion of the PVR. Completion of the PVR establishes the CPCI product baseline and initiates the formal test phase as described in Chapter 6.

4-9. SYSTEM VALIDATION REVIEW (SVR). The purpose of the SVR is to review the results of validation testing to insure that the ADS satisfies the requirements of the SS and the FD. The SVR is

supported by the final FCA and PCA.

a. Review items, The following items are reviewed at the SVR.

(1) PVR minutes

(2) Findings and recommendations resulting from the final FCA and PCA

(3) Test Report (RT)

(4) Technical documentation as required

(5) Configuration management records as required

b. A prerequisite for successful completion of the SVR is the certification by the functional OPR or designated functional representative that the ADS adequately satisfies the requirements stated in the FD. Completion of the SVR terminates the test phase and initiates the operation phase.

AFR 300-15

AUTOMATED DATA SYSTEM PROJECT MANAGEMENT

June 1977 (Draft)

Attachment 1

DEPARTMENT OF THE AIR FORCE AF REGULATION 300-15

Headquarters US Air Force

Washington, DC 20330

Data Automation

AUTOMATED DATA SYSTEM PROJECT MANAGEMENT

This regulation outlines policies and procedures which provide a methodology for the management of data system (ADS) projects. It must be used with AFRs 300-2, 300-7, and 300-12. It applies to all Air Force activities responsible for planning, designing, developing, authorizing, selecting, acquiring, maintaining, and managing ADS projects. It implements DOD Manual 4120.17M, December 29, 1972 and AFR 65-3, July 1, 1974.

OPR: AF/KRAX (Maj M.D. Hall)

KRA APPROVAL: Brig Gen Frederick L. Maloy

WRITER EDITOR:

DISTRIBUTION: F

JUN 77

TABLE OF CONTENTS

Page

FIGURES:

Chapter 1

INTRODUCTION

1-1. PURPOSE: To prescribe a methodology for the management of automated data system (ADS) projects. This regulation provides guidelines for organizing, planning developing and maintaining an ADS using the engineering disciplines of baseline configuration management, reviews and audits, and quality assurance.

1-2. TERMS: Terms used in this regulation are defined in AFR 300-2, AFR 300-12, Volume I, and Attachment 1, this regulation. The heirarchy of terms used in this regulation to describe computer programs are Automated Data System (ADS), Computer Program Configuration Item (CPCI), and module. The relationship of ADS and CPCI is described in paragraph 3-3a.

1-3. OBJECTIVE: To provide guidance for the management of ADS projects.

1-4. APPLICABILITY: The principles and concepts of configuration management apply to ADSs developed under the 300-series Air Force Regulations. The following guidance is provided to aid in the use of this regulation.

a. The disciplines described are applicable to all ADS projects. The procedures will be applied in their entirety to those systems which require more than 50 man-years development effort. For systems requiring 50 man-years or less, the degree to which the procedures apply will be consistent with the system's cost and complexity.

b. Utilize Figure 1-1 to determine the documentation and the reviews and audits for ADS projects.

c. A single, common set of configuration management procedures and quality assurance procedures may not meet the needs of all Air Force organizations. These procedures should be tailored to recognize variations in requirements, organizations, and working relationships. MAJCOM/SOA supplements to this regulation will be prepared if required.

d. Any organizational alignments which may be implied in this regulation are not mandatory.

e. The project manager will establish the degree of formality of reviews and audits consistent with the size of the project.

DOCUMENTATION AND REVIEW DETERMINATION

DEVELOPMENT MAN-YEARS	ALL USAF STANDARD ADS AND MULTISITE COMMAND-UNIQUE ADS	SINGLE-SITE UNIQUE ADS
L.T. 1	UM, OM, MM	UM, MM.
G.E.1 L.T. 10	SRR, SDR, SVR FD, SS, UM, OM, MM	SVR UM, OM, MM
G.E. 10 L.T. 25	SRR, SDR, PVR, SVR FD, SS, PT, UM, OM, MM	SRR, SDR, SVR FD, SS, PT, UM, OM, MM
G.E. 25 L.E. 50	SRR, SDR, PDR, CDR, PVR, SVR FD, SS, PS, PT, RT, UM, OM, MM	SRR, SDR, CDR, SVR FD, SS, PS, PT, UM, OM, MM
G.T. 50	SRR, SDR, PDR, CDR, PVR, PCA, FCA, SVR FD, SS, PS, PT, RT, UM, OM, MM	SRR, SDR, PDR, CDR, PVR, PCA, FCA, SVR FD, SS, PS, PT, RT, OM, UM, MM

NOTES:
1. See Attachment 1 for definition of acronyms.
2. L.T. = less than
   G.E. = greater than or equal to
   L.E. = less than or equal to
   G.T. = greater than
3. RD and DS will be developed as required
4. Above guidelines are minimums; additional requirements may be levied via the data project directive.
5. The above guidelines apply to modifications except that documentation requirements will consist of updates to existing documentation.
6. Computer programs for one-time reports are exempted.

FIGURE 1-1

Chapter 2

OVERVIEW OF ADS PROJECT MANAGEMENT

2-1. PURPOSE. To introduce and briefly define the documents that are prepared, the project management functions that are performed, the development discipline that is followed, and the reviews and audits necessary to insure the successful application of baseline configuration management techniques to ADS projects. (See Figure 2-1.)

2-2. CONCEPTUAL PHASE. A determination of mission and system requirements. This regulation describes those ADS development events which occur after the conceptual phase and formal approval of the Data Automation Requirement (DAR). Procedures for initial analysis, program adversary/advocacy actions, requirements approval and management planning are provided by AFRs 300-2, 300-12, Volume I, and 300-7.

a. TASKS.

(1) IDENTIFICATION OF A REQUIREMENT. A functional manager's description and justification of a "need" for a capability to meet a mission or operational requirement is the beginning of the conceptual phase. The user must inject his expertise into development of a descriptive definition of the initial system concept.

(2) REQUIREMENTS ANALYSIS. Requirements analysis is the identification and definition of the problems associated with providing, changing, or converting a management or

operational capability that will best meet the mission of an organization. The set of requirements that emerges from a requirements analysis is the main device used to achieve project direction and control. The inability to produce this governing set of requirements may be an indicator of a failing project. Thus, it is essential that:

(a) Requirements analysis be thorough and avoid specifying a design solution.

(b) Requirements analysis documentation contain a complete and understandable definition of the requirements.

(c) Compliance with regulatory requirements dealing with policies and procedures be enforced.

(3) PREPARATION OF DOCUMENTATION. Once the user has identified a requirement, evaluated alternative solutions, documented requirements for management evaluation, and automation is selected as the solution, certain documents must be prepared.

b. DOCUMENTS. Documents prepared during the conceptual phase include the Data Automation Requirement (DAR), the Data Project Directive (DPD), the Data Project Plan (DPP), and the Functional Description (FD). The DAR, DPD, DPP, and FD preparation requirements are specified in AFR 300-12, Volume I.

c. SYSTEM/SUBSYSTEM REQUIREMENTS REVIEW (SRR). The SRR is conducted during the conceptual phase to assure the user

that the project's progress is responsive to the approved requirement and to determine initial direction of the design effort (paragraph 4-3).

2-3. DEFINITION PHASE. In this phase one or more system developers define the design requirements for the major elements in the system.

a. TASKS.

(1) DEVELOP SYSTEM INTERFACE CONTROL REQUIREMENTS. Define interface between operational functions. Include:

(a) Initial performance allocation.

(b) Sequencing/scheduling.

(c) Control techniques.

(2) EXPAND SYSTEM REQUIREMENTS. Perform a comprehensive and critical review of functions, performance, and design requirements, and expand them.

(3) DEFINE COMPUTER PROGRAM REQUIREMENTS. Define requirements to be satisfied by the operational computer programs. This definition should include both narrative and graphic presentation of information flow.

(4) DEFINE EQUIPMENT PERFORMANCE REQUIREMENTS. Define requirements for the tasks to be performed by the computer and other system hardware.

(5) PREPARE DOCUMENTS.

b. DOCUMENTS.

(1) SYSTEM/SUBSYSTEM SPECIFICATIONS (SS). The SS is a technical document that governs the development of an ADS

or subsystem of an ADS. System/Subsystem Specifications define performance, interfaces, and other technical requirements sufficiently to permit detailed design. Approval of this document at the System Design Review (SDR) establishes the allocated baseline for the system or subsystem.

(2) DATA REQUIREMENTS DOCUMENT (RD). A document that specifies the characteristics and limitations of required data. Defines inputs required of the user, procedures for providing this input to the system files, expected outputs, use of standard data elements and codes, and data limitations.

(3) DATA BASE SPECIFICATION (DS). This document specifies the design of the data base and establishes the definitions of the interfaces.

c. SYSTEM DESIGN REVIEW. The SDR is a review of the total system requirements used to produce the system definition (system/subsystem specifications). Successful completion terminates the definition phase (paragraph 4-4).

2-4. DEVELOPMENT PHASE. Analysis and design, coding, debugging, integration, and development testing of computer program configuration items (CPCI) is done in this phase.

a. TASKS. The sequence of tasks during the development phase is:

(1) REVIEW SYSTEM/SUBSYSTEM SPECIFICATIONS (SS). Perform a review of the subsystem design. When the ADS is defined as a single CPCI, the System Design Review (SDR) and Preliminary Design Review (PDR) can be combined.

(2) PRELIMINARY PROGRAM DESIGN.

(a) Define functions which will be performed by individual programs.

(b) Complete definition of inputs, processing, and outputs of programming tasks. When defined, assign to specific modules and develop refined functional flows. A system simulation study may be necessary to avoid the risk of serious design deficiencies.

(3) INITIATE SUPPORT DOCUMENTATION. Begin preliminary drafts of the Program Maintenance Manual (MM), Users Manual (UM), Computer Operation Manual (OM), and Test Plan (PT).

(4) DETAIL FUNCTION DESIGN. Expand flowcharts, hierarchical input/process/output (HIPO) charts, structure charts, or other logic designs, algorithms, and narrative descriptions in sufficient detail to provide the basis for actual coding. Complete definition of the data base at the module level, including number, type, and structure of tables and description of items in the tables. Define all elements as dynamic or static, and their uses.

(5) CONDUCT CRITICAL DESIGN REVIEW. Conduct a user and developer review to confirm that the design meets its functional development requirements, and that the design is sufficiently defined to permit the start of coding. Approval of program specifications (PS) at the Critical Design Review (CDR) finalizes the design of the ADS.

(6) CODE, COMPILE, AND VERIFY. Code, compile and verify modules. Conduct development testing of coded modules until complete programs and systems are developed and verified. Development programmers are responsible for development testing.

b. DOCUMENTS. Documents initiated in previous phases are updated, and the following documents are initiated.

(1) TEST PLAN (PT). A document that specifies the test conditions for acceptance testing of a computer program. The test plan is based on the requirements of the FD and the SS.

(2) USERS MANUAL (UM). Describes in general, non-technical terms, the applications of the system and its computer programs, and provides specific information about inputs and outputs to assist the user in effectively utilizing the system.

(3) COMPUTER OPERATION MANUAL (OM). Contains the information necessary to operate the ADS, including deck setups, inputs required, operating instructions, error messages, and recovery procedures.

(4) DEVELOPMENT TEST PLAN. A document that specifies method and content for development testing. Defines test management, reports, controls, manpower, acceptance criteria, and test procedures (Attachment 2).

(5) PROGRAM MAINTENANCE MANUAL (MM). A document maintained by a programmer or programmer team for each

module for which responsibility is assigned. It includes requirements, interfaces, design, and test information (Attachment 3).

(6) OTHER PLANS. Detailed supporting plans required as annexes to the Data Project Plan (see AFR 300-12, Volume I) are normally prepared during the development phase. Such plans address training, communications, site preparation, etc.

c. REVIEWS. The following reviews are conducted during the development phase.

(1) PRELIMINARY DESIGN REVIEW (PDR). The PDR is a review of the subsystem design approach for a CPCI occurring prior to the start of detailed design (paragraph 4-5).

(2) CRITICAL DESIGN REVIEW (CDR). The CDR is a review conducted on each module or a set of modules within a CPCI before translating the detailed design to coded instructions (paragraph 4-6).

(3) CONFIGURATION AUDITS. Conduct functional configuration audits (FCA) and physical configuration audits (PCA) as required to insure software products and documentation comply with project requirements. These audit functions are started prior to the PVR and are updated prior to the SVR (paragraph 4-7).

(4) PRODUCT VERIFICATION REVIEW (PVR). The PVR is a review of each CPCI to establish the product baseline for

that CPCI and to insure preparation for the Test Phase has been completed (paragraph 4-8).

2-5. TEST PHASE. The installation and validation test of the system in the operational environment by the designated user representative or by an independent agency.

a. VALIDATION TESTING. Conduct two-stage validation testing of the system by an independent test group. In the first stage, perform in-house validation testing of the system requirements, correct problems, and update documentation. In the second stage, rerun validation test for the user, or review successful in-house validation test runs with the user. In addition, validate program specifications and other relevant documentation in both stages (paragraph 6-5).

b. SYSTEM VALIDATION REVIEW (SVR). The SVR validates that performance of a CPCI, as determined through validation testing, complies with the SS and the FD (paragraph 4-9).

2-6. OPERATION PHASE. The operation, maintenance, and product improvement by the user and/or the developer.

2-7. PROJECT MANAGEMENT FUNCTIONS. The organizational structure selected to develop an ADS project is dictated to a certain degree by available manpower, the project's scope and complexity, schedule requirements, and other project-

unique factors. The project manager is ultimately responsible for insuring adherence to management policies and practices. He must plan major functions of the project and insure that appropriate planning is being conducted at lower levels. Primary management functions are briefly introduced here, and covered in detail in later chapters.

a. QUALITY ASSURANCE. Quality assurance consists of all actions that are taken to assure the development organization delivers products that meet performance requirements and adhere to standards and procedures. (See Chapter 5.)

b. 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. (See Chapter 3.)

c. TESTING. Testing verifies that software products conform to performance and design requirements. Test plans and procedures must be prepared, test data made available, test facilities provided, and schedules must include adequate test time. (See Chapter 6.)

e. PLANNING AND CONTROL. The planning and control function includes preparation of budget estimates and tracking of actual costs; tracking progress against project schedules; monitoring reviews and audits; and preparation of the documentation plan.

f. SOFTWARE DEVELOPMENT. Software development includes design, coding, and development testing of products; generation of materials for configuration management reviews and audits; and generation of all technical documentation.

g. OPERATIONAL SUPPORT AND MAINTENANCE. The operational support and maintenance function includes updating computer operation manuals (OM), user manuals (UM), and program maintenance manuals (MM); support of system testing; review of new software requirements and design; correction of software problems; and improving and modifying existing software.

Chapter 3

CONFIGURATION MANAGEMENT

3-1. PURPOSE. This chapter provides detailed procedures for software configuration management, from the conceptual through the operation phases of a system. It provides a general software configuration management approach which can be tailored to specific project requirements. The key to configuration management is knowing what you have and keeping track of changes.

3-2. CONFIGURATION MANAGEMENT PLAN (CMP). Software configuration management must be responsive to user requirements. Complex software projects require a highly organized configuration management (CM) activity to insure project objectives are achieved; whereas, less complex software projects may require nothing more than control of applicable specifications and acceptance testing of the products. A CMP is part of the Data Project Plan and describes project responsibilities and procedures for implementing CM, and includes schedules for major CM activities and events for life cycle of the system. The CMP should be developed and approved prior to start of the definition phase (Attachment 4).

3-3. CONFIGURATION IDENTIFICATION. Underlying all other aspects of software configuration management is a progressive and increasingly detailed identification of the computer program configuration by means of baseline documentation. The required configuration of a computer program configuration item (CPCI) is identified by its system/subsystem

specification (SS), and its achieved configuration by its program specification (PS). Other documents that may be part of a CPCI's configuration identification are the data requirements document (RD), if it is referenced in the SS, and the data base specification (DS), if it is referenced in the PS. Other documents are not part of a CPCI's configuration identification. They are influenced by, but do not identify or control a CPCI's configuration identification.

a. DESIGNATING A CPCI. The term CPCI is used to define the product or products subject to the application of configuration management procedures. A CPCI may be an entire ADS or portion thereof. This determination is basically a management decision. Trade-offs are based on judgement and experience. Factors bearing on the decision include the risk associated with the ADS, its planned use (single or multiple users), its complexity, the level of change control required by management, the modification activity expected during the life of the ADS, and cost thresholds imposed by management based on the estimated value of the ADS. For example, an ADS may consist of a data base management system and several application programs. The ADS may be defined as a single CPCI or each logical component may be defined as a CPCI.

b. SELECTION OF PROJECT DOCUMENTS. To determine the items needed in a project documentation set, a documentation analysis should be performed during the definition phase. Such an analysis will help to insure that project documentation adequately covers all information needed with a minimum

of overlap or duplication.

c. PREPARING PROJECT DOCUMENTATION. The development activity is normally responsible for preparing and delivering the required documentation. Documentation responsibilities should be prepared specifying the individuals that are responsible for preparing and coordinating each document from origination to formal release and updating.

(1) Each project manager should insure that:

(a) Documentation requirements are adequately and appropriately identified for the project.

(b) A documentation plan is established as early as possible in the ADS life cycle.

(c) Project documentation requirements are understood by personnel responsible for documentation preparation, coordination, distribution and updating.

(2) The personnel preparing each document should be responsible for:

(a) Thoroughly understanding the project documentation requirements as specified in the documentation plan.

(b) Understanding the intended use of the document and the requirements of the intended users.

(c) Understanding the overall content and appropriate level of detail required for the document.

(d) Complying with applicable ADP standards for document format and content.

d. PREPARING A SOFTWARE DOCUMENTATION PLAN. A documentation plan is first developed in draft form and usually refined during the definition phase of the system. Essential steps in preparing a software documentation plan are:

(1) Establish the scope and general outline of the documentation plan. Generally the documentation plan covers technical documentation, management documentation, reports and official letters, and standard forms to be used on the project.

(2) Define the specific document types that have been selected for the project. For each such document state:

(a) Title and identification number.

(b) Purpose.

(c) Responsibilities and general procedure for organization and updating.

(d) Coordination and approval authorities.

(3) Schedule each document to be available at the time its information is required for review, or release, based on the projects particular need.

(4) Define responsibilities and procedures for publication and distribution of documents.

e. DOCUMENTATION STANDARDS.

(1) Documentation for ADS development will be prepared in accordance with DOD Manual 4120.17-M, AFR 300-12, Volume I, and this regulation.

(2) Technical documentation may be combined under one cover with each type designated as a separate part; e.g., Part I, Part II, etc., and the format standard for each type still followed.

f. COMPUTER PROGRAM LIBRARIES. Computer program libraries, also called programming support libraries, are automated data repositories that have been successfully used as an aid in ADS development projects to the actual software development process and the management of the project. Project managers should consider the use computer program labraries early in the definition phase.

(1) Support of the software development process includes:

(a) Storage and maintenance of computer programs and related data.

(b) Compilation and testing of computer programs.

(c) Generation of documentation.

(2) Support to project management includes:

(a) Collection and reporting of management data related to software development.

(b) Control over the integrity and security of data stored in the library.

3-4. CONFIGURATION CONTROL. Configuration control is the systematic evaluation, coordination, approval or disapproval, and implementation of all approved changes in the

configuration of a CPCI after formal establishment of its configuration identification. This process assures adequate analysis of changes that affect cost, schedule or design, and provides controlled updating of evolving software configuration throughout the ADS life cycle. This paragraph describes the concepts, forms, and procedures for achieving the proper degree of configuration control during all phases of a project.

a. CHANGE CLASSIFICATIONS. Changes to software products or documentation can be placed into three classifications each requiring different processing and review procedures depending on the impact of the change.

(1) CLASS I CHANGES. Class I changes are changes which impact approved schedules, cost, or baselines. Class I changes require approval by the configuration control board (CCB) and/or the appropriate ADP approval authority if the change exceeds the authority delegated to the project manager. Generally they include the following kinds of changes;

(a) Technical changes to the functional, allocated, or product baselines of the CPCI.

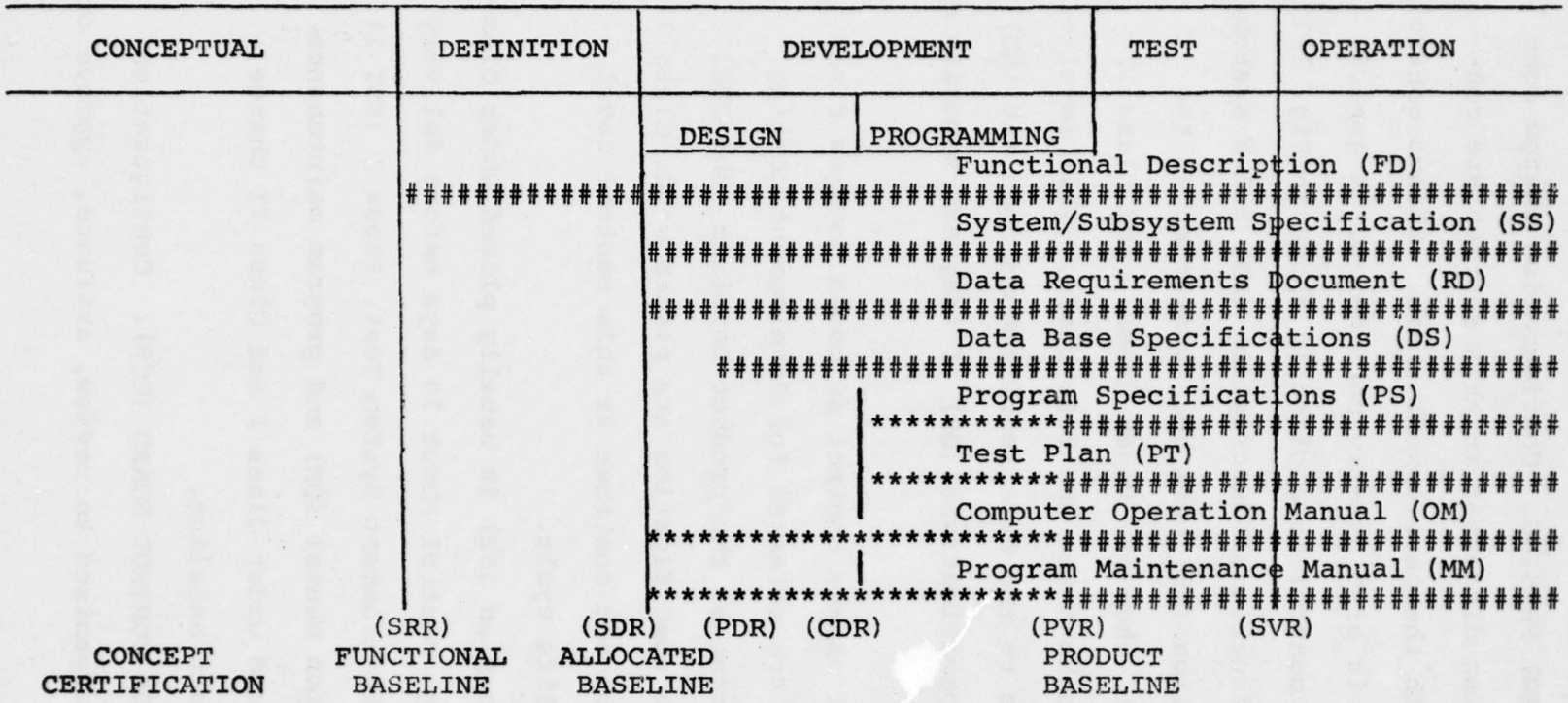
(b) Changes involving performance outside the specified tolerances of the CPCI or affecting interface characteristics with other CPCI's.

(2) CLASS II CHANGES. Class II changes are changes affecting documentation only, or changes which are classified

as maintenance. Class II changes do not require configuration control board (CCB) approval prior to implementation, but copies of Class II changes must be forwarded to the project manager (or his designee) for concurrence in the change classification.

(3) INTERNAL CHANGES. Internal changes are changes to a CPCI, or document that has not been released for Class I and Class II change control. For example, changes to a preliminary program specification (PS) prior to the product baseline are treated as internal changes. Similarly, any changes required during development testing are treated as internal changes unless the change affects the functional or allocated baselines, costs, or schedules.

CONFIGURATION CHANGE CONTROL PHASING DURING PROJECT LIFETIME



LEGEND

######## Class I and II Control

******** Internal Change Control

Figure 3-1

b. CHANGE CONTROL PHASING. Configuration change control tasks and phasing discussed in this section are concerned primarily with the development phase. Change control during other phases is also briefly considered. A general view of when change control is applied is shown in Fig. 3-1.

(1) When a functional description (FD) and a system/subsystem specification (SS) have been baselined at the start of development, they are under Class I and Class II change control throughout the ADS life cycle. This level of control also applies to the data requirements document (RD) and the data base specifications (DS) if they are separate documents.

(2) Internal change control of coded modules first occurs when modules are released for development testing. Following establishment of the product baseline, the CPCI and the CPCI program specification are placed under Class I and II change control and continue at this control level throughout the ADS life cycle.

(3) The test plan (PT) is usually placed under Class I and Class II change control about 30 days before delivery of the product for Environment System Test, Phase I (EST I). The computer operation manual (OM) and program maintenance manual (MM) are placed under Class I and Class II change control at the product baseline.

c. CONFIGURATION CONTROL BOARD (CCB). Configuration control boards are organized to review, evaluate, approve or

disapprove, and release all Class I changes. As previously indicated, Class I changes may require approval of both the CCB and the appropriate ADP approval authority. The CCB must approve Class I changes before they are submitted (if required) to the appropriate ADP approval authority.

(1) The CCB usually is chaired by the project manager. The members are the user representative, development and quality assurance personnel. Attendance at meetings depends on impact of the change request being reviewed.

(2) The CCB is organized at the start of the project. Policies and procedures required for the operation of the CCB will be prepared and implemented. A secretary will be appointed with responsibility for preparing the agenda, agenda items, and meeting minutes.

d. CHANGE CONTROL FORMS. Change control forms are the basic media for reporting problems, and processing changes. They are the key source of information concerning the status of changes during change processing. Changes to baseline documentation must be controlled using the Baseline Change Request (BCR), or equivalent form, paragraph d(3). With the exception of the BCR, use of the following forms is optional. Sample forms and instructions for preparation are contained in Attachment 6.

(1) SOFTWARE PROBLEM REPORT (SPR). An SPR is used after initiation of development testing to report a known or suspected discrepancy or deficiency in an existing computer

program, in its documentation, or in interfacing software. The SPR provides a record of the discrepancy from initiation to problem resolution.

(2) DATA BASE CHANGE REQUEST (DBCR). A DBCR is used to control changes to a data base after the data base in placed under configuration control. When a problem is reported via an SPR and a solution to the problem requires a data base change, a DBCR is prepared and distributed. After approval of the DBCR, the data base is updated accordingly.

(3) BASELINE CHANGE REQUEST (BCR). A BCR or equivalent form is used to propose Class I and Class II changes. It provides a description of the proposed change and the impact of the change on the cost, schedule, and performance of the CPCI.

(4) DESIGN PROBLEM REPORT (DPR). The DPR provides a standard means for documenting problems identified during formal reviews and audits.

(5) REQUEST FOR DEVIATION/WAIVER. A request for deviation or waiver is used by a developer to request and document temporary departures from configuration identification requirements when permanent changes are not acceptable. A deviation is a written authorization granted prior to product development to permit a developer to depart from a particular performance or design requirement for a specified product or period of time. A waiver is a written authorization to deliver a configuration item that has been found

after development to depart from specified requirements but that nevertheless is considered suitable for use or rework. A BCR may result in a deviation or waiver.

e. CHANGE PROCESSING. A change may be requested by any organization associated with the project. All change requests initiated in a project are first reviewed at the lowest approval level in the project. If the change does not affect approved baselines, action should be implemented at the lowest level. Change processing guidelines are:

(1) CLASS I CHANGE CONTROL. Class I changes require a BCR to be submitted for approval to the configuration control board (CCB). If the change exceeds delegated approval thresholds, the BCR is used to prepare an amended DAR which is submitted to the appropriate ADP approval authority. Instructions for submitting an amended DAR are contained in AFR 300-12, Volume I. Implementation of most Class I changes follows an independent development cycle until it can be merged with the current software system. After start of the operation phase, all modifications are processed as DARs.

(2) CLASS II CHANGE CONTROL.

(a) Prior to the operation phase, Class II changes must be approved at the level designated by the project manager. Concurrence of proper classification should be obtained from the project manager (or his designee).

(b) During the operation phase, Class II changes must be approved by the ADS manager or his designee.

(c) Release of a Class II change is authorization to implement the change.

(3) INTERNAL CHANGE CONTROL.

(a) Internal changes do not require CCB approval or concurrence.

(b) Modules verified or updated during development testing are included in the project library and integrated into the CPCI. In certain real-time and on-line applications programs may be patched to conserve limited test time and maintain development schedules. All "patches" must be documented in a log and entered as changes into the CM system.

f. MULTIVERSION CPCIs

(1) More than one version of a CPCI may exist with each version in a different phase of the ADS life cycle. Multiple versions can occur under the following circumstances.

(a) Follow-on development to an ADS currently in its operation phase. Changes which are not of an emergency nature should be collected and implemented as a single release when feasible. They are usualy referred to as "block releases".

(b) A product developed in two phases, each independently operational, such as initial operational capability (IOC) and full operational capability (FOC).

(c) An addition to a CPCI currently in development. For example, a BCR is issued to expand the capability

of a product currently at the product baseline, with the new version to be started at the allocated baseline.

(d) A product being developed incrementally to utilize manpower at a steady rate.

(2) A major consideration in controlling multiple versions is the coordination of changes. For example, version "A", currently in system testing, has functional, allocated, and product baselines, while version "B", in development testing, has only functional and allocated baselines. A change of small consequence to "B" might have severe implications on "A". Therefore, all versions must be considered whenever a change is incorporated in any of them. To insure that such consideration has been given, changes should be logged against all effected versions.

g. MULTIPLE CONFIGURATION AND MULTIPLE SITE PROJECTS.

(1) Multiple configurations may also exist and be controlled at the same time. This situation occurs when an ADS is installed or about to be installed at different locations, and each location is to maintain its own configuration. Each location's ADS is treated as a separate CPCI, and problems at any location should be submitted to other locations to see if there is any impact.

(2) When a single configuration is required for all locations, the configuration is controlled from one location. The other locations maintain a record of any temporary deviations from the configuration.

h. SMALL PROJECTS. On projects less than one man-year of development effort where constant informal contact among all participants is possible, there may be no need for formal change control procedures. However, program maintenance manuals should be maintained to document the development process, including problems, corrective action, and current configurations.

3-5. CONFIGURATION STATUS ACCOUNTING. Configuration status accounting is the process of recording and reporting the status of an evolving CPCI throughout the ADS life cycle. Status accounting records include a log for each change control form, with sufficient cross referencing between logs to facilitate tracking and preparation of reports. The logs may also be employed for analyses of development and test activities. Attachment 5 contains a brief description of logs which may be used for configuration status accounting. The following documents may be used to report information regarding the ADS.

a. TURN-OVER (TRANSMITTAL) LETTER. The turn-over (transmittal) letter identifies all products delivered and cross-references applicable documentation. The turn over letter contains the following information as a minimum:

(1) INVENTORY OF MATERIALS RELEASED. A list of all items (tapes, cards, disks, etc.) forwarded with the release. All utility and/or support computer programs, not part of the release, but required to operate, load, or regenerate the release are identified.

(2) ADAPTATION DATA. For the release of a new CPCI version, identify (by reference to appropriate specifications and/or listings) all unique-to-site data in the items being released.

(3) INTERFACE COMPATIBILITY. For the release of a new CPCI version, indicate other systems and/or CPCIs affected by the changes incorporated in this release.

(4) BIBLIOGRAPHY OF REFERENCE DOCUMENTS. This section is included only for the release of a new version of a CPCI and lists all pertinent documents related to the released version.

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

(6) PROBLEMS AND RESTRICTIONS. List all known problems associated with the delivered CPCI and any restrictions pertaining to its use.

b. PRODUCT STATUS REPORTS.

c. COMPUTER PROGRAM CONFIGURATION ITEM (CPCI) INDEX.

(1) A CPCI index describes the current status of all baseline specifications and support documents pertaining to a CPCI. Document status is given in terms of issue dates, document numbers and titles, BCRs and revision identifiers associated with each document issue or revision resulting from implemented changes. The index also contains a development record that gives completion dates for the key

events of CPCI development.

(2) A CPCI index is initially issued within 30 days of the establishment of the allocated baseline for each CPCI and is updated as required. Each succeeding issue is expanded and revised to reflect progress through the key events.

Chapter 4

REVIEWS AND AUDITS

4-1. PURPOSE. To describe the reviews and audits that are scheduled at meaningful points during the ADS life cycle.

4-2. PARTICIPATION AND RESPONSIBILITIES. The project manager or his designee chairs all reviews and audits. The user representative will make sure that the functional requirements are being adequately met.

a. Project Manager Responsibilities.

(1) Establish the time, the place, and the agenda for the reviews and audits, and coordinate these arrangements with all participants. Depending on the type of review or audit, schedules for incremental reviews may have to be prepared and coordinated with all participants.

(2) Prepare for each review or audit in sufficient depth consistent with project scope and size. Material to be reviewed must be submitted to participants prior to each review, allowing time for comment. Documents should be reviewed for technical rather than editorial content. Review participants should prepare design problem reports (DPRs) and submit them to the originator of the material being reviewed. The originators of the material will evaluate the DPRs and reply to them at the formal review. Another approach is to correct minor discrepancies prior to the review, advise participants of the disposition, and hold the major DPRs for discussion at the meeting. After the

review, the documentation is updated, incorporating necessary changes, and formally released.

(4) Provide a secretary to record official meeting minutes. Minutes are recorded only as directed by the chairman or the user representative and consist of significant questions and answers, action items, deviations, conclusions, and recommended courses of action resulting from presentations or discussions.

(5) Publish and distribute official minutes of the meeting.

(6) After each review or audit, provide the review participants with the development activity's position on each action item in the minutes.

b. User Responsibilities.

(1) Coordinate the arrangements for the review or audit with the project manager.

(2) Review all documentation provided by the development activity prior to the review or audit to validate that user requirements are stated properly and have been adequately translated into the ADS design. Discrepancies should be documented by DPR and submitted to the development activity.

(3) Within 30 days following each review or audit, provide review participants with a position on each action item in the minutes.

c. Resolution of disagreements between the development activity and the user will be escalated to the next level of management for resolution.

4-3. SYSTEM/SUBSYSTEM REQUIREMENTS REVIEW (SRR). The SRR is conducted during the conceptual phase to assure the user that the project's progress is responsive to the approved requirements, and to determine initial direction of the design effort.

a. Review Items. The principal items to be reviewed at the SRR are the Data Automation Requirement (DAR), the Data Project Directive (DPD), the Data Project Plan (DPP), and the draft Functional Description (FD). A detailed review of mission and operational requirements, preliminary requirements allocation, system interface studies, etc., should be made at this time.

b. Postreview Action. After completion of the SRR, the minutes are distributed by the project manager. Notification of any required postreview action items, plus official acknowledgment of conduct and completion of the review, are provided to the user, the developer, and the appropriate ADP approval authority. Approval of the FD at the SRR establishes the Functional Baseline and initiates configuration management of the FD.

4-4. SYSTEM DESIGN REVIEW (SDR). The SDR establishes the allocated baseline for individual CPCIs. Upon approval of

the system/subsystem specification (SS), configuration management of the SS is initiated.

a. Review Items.

(1) The SDR documentation package should include the following:

(a) Functional Description (FD) with changes.

(b) System/subsystem specifications (SS).

(c) Tradeoff studies, analyses, etc.

(d) Data Project Plan (DPP).

(2) The SDR is conducted to accomplish:

(a) Review any changes to the FD.

(b) Insure the allocated requirements implement the functional requirement stated in the FD.

(c) Insure the technical project risks are identified, ranked, avoided, and adequate tradeoffs are made.

(d) Insure a technical understanding of requirements has been reached and technical direction is provided to project participants.

(e) Insure project schedules and milestones are consistent with the allocated requirements.

b. Postreview Action. After completion of the SDR, the minutes of the meeting are coordinated with all participants. Participants will also be advised of any postreview action items requiring response. Approval of the the SS establishes the allocated baseline.

4-5. PRELIMINARY DESIGN REVIEW (PDR). The PDR is a technical review of the subsystem design for a computer program configuration item (CPCI). The PDR for each CPCI occurs between SDR and CDR in the development phase. Only one successful PDR per CPCI is required. A collective PDR for a functionally related group of CPCIs, treating each CPCI individually, may be held when advantageous to the project. The overall technical risks associated with each CPCI are also reviewed on a technical, cost, and schedule basis. NOTE: If the ADS is defined as a single CPCI, the PDR may be combined with the SDR.

a. Review Items.

(1) The PDR documentation package should include the following.

(a) Updated functional description (FD).

(b) System/subsystem specification (SS), including approved changes.

(c) Draft of CPCI test plans (PT), less test procedures.

(d) Data base structure and organization.

(e) Design problem reports (DPR).

(2) The PDR is conducted to establish the following:

(a) Compatibility of the design with the SS for the CPCI and with other CPCIs with which it must interface.

(b) Adequacy of test plan.

(3) To satisfy the intent of the PDR, the following tasks should be performed.

(a) Review all functional interfaces. Review message formats, storage availability, and other considerations that may have been established in the SS. At this time, the interfaces between CPCI and hardware CIs should be defined at a level low enough to satisfy all software development needs.

(b) Review the structure of the CPCI as a whole, with emphasis on allocation of the functions delineated in the SS to individual computer programs; storage requirements and allocation; computer program operating sequences; and design of the data base.

(c) Analyze critical timing requirements to insure the proposed CPCI design satisfies the timing requirements.

(d) Review human factors impact of proposed subsystem design of the CPCI.

(e) Review all changes to the SS subsequent to the established allocated baseline. Insure the FD and SS adequately reflect these changes.

(f) Review the PT to insure it satisfies the requirements of the FD and SS.

(g) Review status of all negative and provisional entries such as "not applicable" or "to be determined" in the SS.

b. Postreview Action. After completion of the PDR, the minutes are published and distributed by the development activity. The user will be notified of any required post-review action.

4-6. CRITICAL DESIGN REVIEW (CDR).

a. Purpose. The CDR is a review conducted during the development phase before translating logic, and algorithms to coded instructions. The CDR insures the detailed design solution, as reflected in the draft program specification (PS), satisfies performance requirements established by the SS. The CDR also verifies system design compatibility with other CPCIs, and within the CPCI. Approval of the project manager is required to release portions of CPCIs for coding prior to CDR when necessary to maintain schedules.

b. Review Items and Tasks

(1) The CDR documentation should consist of the following:

(a) System/subsystem specification (SS), including approved changes.

(b) CPCI test plans (PT), including test procedures.

(c) Design problem reports (DPR).

(d) Preliminary design review (PDR) minutes.

(e) Draft of computer operation manual (OM).

(f) Draft of program maintenance manual (MM).

(g) Draft of the program specification (PS).

(h) Draft of users manual (UM).

(i) Appropriate support documentation, such as updated timing studies, accuracy studies, etc.

(2) As a minimum, the following tasks are to be performed:

(a) Determine compatibility of the PS with the SS.

(b) Review all interfaces between CPCIs and within a CPCI.

(c) Review interfaces between CPCI and the applicable hardware CIs to insure changes have not affected compatibility.

(d) Review the PT for technical adequacy and compatibility with the FD and SS requirements.

c. Postreview Action. After completion of the CDR, the minutes are published and distributed. Participants will be notified of any required postreview action. Successful completion of the CDR permits commencement of programming.

4-7. CONFIGURATION AUDITS.

a. Purpose. A configuration audit is a process employed to verify conformance of a CPCI to specifications and standards. Two audits, the Functional Configuration Audit (FCA) and the Physical Configuration Audit (PCA), are required at two points during the ADS life cycle for an ADS requiring greater than fifty development manyears. The first point, prior to the Product Verification Review (PVR), is termed

the preliminary FCA and PCA, and the second point is prior to the System Validation Review (SVR) and termed the final FCA and PCA. The preliminary audits are a thorough technical examination of the CPCI, while the final audits are directed at changes made to the CPCI as a result of formal testing. The following guidelines for conducting the FCA and PCA are provided.

b. FUNCTIONAL CONFIGURATION AUDIT (FCA). The FCA validates the CPCIs actual performance compliance with the SS. Test data is reviewed to validate that the CPCI performs as required. The project manager provides two FCA data packages to the organization performing the FCA, one to be delivered at some negotiated lead time prior to FCA, and the other to be made available during the FCA.

(1) Items to be provided before audit are:

(a) Nomenclature (name or descriptive title of the CPCI).

(b) Identification of necessary documentation.

(2) The documentation package to be made available for FCA consists of:

(a) Program specification (PS), including listings.

(b) System/subsystem specification (SS) with changes, if applicable.

(c) Test plan (PT) and when available, test reports (RT).

(d) PDR and CDR minutes.

(e) Status accounting records of SPRs, DBCRs, and BCRs.

(3) The FCA consists of examining and evaluating:

(a) Test procedures and results to determine CPCI conformance with the SS and FD and quality assurance provisions.

(b) Adequacy of analysis or simulations where performance parameters cannot completely be verified by test.

(c) Any requirements stated in the FD that could not be met, and the developer's proposed solution.

(d) All approved BCRs to insure they are incorporated and verified during testing.

(4) Deficiencies noted and recommended corrective actions are documented in the FCA minutes.

c. PHYSICAL CONFIGURATION AUDIT (PCA). The PCA is a formal examination of the coded version of a CPCI against its technical documentation. The PCA includes a selective audit of the program specification (PS), including logic charts, listings, and design narrative. It also includes a review of the format and completeness of any other documentation due for acceptance. The project manager provides two PCA data packages to the organization performing the PCA - one to be delivered at some negotiated lead time prior to PCA, and the other to be made available during PCA.

(1) Items to be delivered prior to audit consist of:

(a) PCA date and location.

(b) Identification of CPCIs to be accepted.

(2) The documentation package to be provided and made available for PCA consists of:

(a) System/subsystem specification (SS).

(b) Program specification (PS).

(c) Test procedures (PT) and test report (RT).

(d) Draft computer operation manual (OM).

(e) Draft program maintenance manual (MM).

(f) CPCI tape and listings.

(g) Applicable configuration management records.

c. The PCA consists of reviewing:

(1) PS for format and completeness.

(2) Selectively compare the listings with documentation.

(3) Computer operation manual (OM) and program maintenance manual (MM) for format and completeness.

(4) The release and change control system to be sure it can properly control the processing and formal release of changes.

4-8 . PRODUCT VERIFICATION REVIEW (PVR).

a. Purpose. PVR is conducted for each CPCI at the end of the development phase to establish the Product Baseline

for that CPCI and to insure preparation for the test phase has been completed.

b. A preliminary functional configuration audit (FCA) and a preliminary physical configuration audit (PCA) will be conducted prior to the PVR to provide management with an independent technical assessment of the CPCI.

c. Items reviewed at the PVR include:

(1) Findings and recommendations resulting from the preliminary FCA and PCA.

(2) Test plans.

(3) Preparations and schedules for the test phase.

d. Discrepancies which would lead to establishing an invalid product baseline and premature testing must be corrected prior to successful completion of the PVR. Completion of the PVR establishes the CPCI product baseline and initiates the formal test phase as described in Chapter 6.

4-9. SYSTEM VALIDATION REVIEW (SVR). The purpose of the SVR is to review the results of validation testing to insure that the ADS satisfies the requirements of the SS and the FD. The SVR is supported by the final FCA and PCA.

a. Review items. The following items are reviewed at the SVR.

(1) PVR minutes.

(2) Findings and recommendations resulting from the final FCA and PCA.

(3) Test Report (RT).

(4) Technical documentation as required.

(5) Configuration management records as required.

b. A prerequisite for successful completion of the SVR is the certification by the functional OPR or designated functional representative that the ADS adequately satisfies the requirements stated in the FD. Completion of the SVR terminates the test phase and initiates the operation phase.

Chapter 5

QUALITY ASSURANCE

5-1. PURPOSE. To describe techniques to improve quality, reliability, and maintainability of an ADS.

5-2. DEFINITION. Quality Assurance (QA) is all actions taken to make sure the development organization delivers products that meet performance requirements and adheres to standards and procedures.

5-3. FUNCTION. The QA function should be independent of the designers and programmers. This function may be performed by an organic organization or by a contractor. The following are primary QA functions:

a. Planning.

b. Development of QA Policies and Procedures.

c. Configuration Audits.

5-4. PLANNING. QA planning starts with a review of the DAR, DPD, DPP, and FD. This review ends in a QA plan which describes the required quality assurance functions, responsibilities, and tools to make sure that software is reliable and can be maintained. The QA plan is coordinated with the project manager.

5-5. POLICIES, PROCEDURES, AND STANDARDS.

a. Performance of QA activities requires establishment of QA policies and procedures. QA policies and procedures outline the rules for conducting configuration audits and assuring compliance with ADP standards requirements and local

procedures for design, development, and test of computer programs, and software deliveries.

b. QA personnel assist in the adaptation and interpretation of directed and local ADP standards. Relaxation from standards may be required due to technical problems and schedule impacts. Deviations from standards must be coordinated with QA personnel.

5-6. CONFIGURATION AUDITS. To make sure QA policies and procedures, and ADP standards are followed, QA personnel conduct the Functional Configuration Audits (FCA) and the Physical Configuration Audits (PCA) as described in Chapter 4. QA personnel will document findings and recommendations.

5-7. QA PARTICIPATION IN REVIEWS. QA personnel should participate in the SRR, SDR, PDR, CDR, and PVR to insure the following:

a. Documentation complies with directed and local standards.

b. Assist other project participants in determining technical adequacy of documentation.

c. Insure project schedules include adequate time for the conduct of configuration audits.

5-8. TEST PHASE SUPPORT. Specific QA tasks to be performed during the test phase are as follows:

a. Make sure master copies of test procedures, test results, and test reports are maintained.

b. Review test results for consistency with test criteria.

c. Monitor all tests to insure actual tests performed are as specified in documented test procedures.

d. Compare the configurations of hardware/software components utilized in the test against the configuration identified in the test procedures.

e. Certify that the test output is correct, the test satisfies requirements, and the test data package is complete.

f. Retain certified test data packages containing a copy of test procedures, test output, and discrepancy reports.

g. Insure compliance with approved quality, test and configuration management procedures.

Chapter 6

TEST MANAGEMENT

6-1. GENERAL. Methods and procedures for testing ADSs are described in this chapter. Details of the testing approach include test validation methods, test control, and test reporting. In addition, testing aspects that affect the user and other organizations are identified. While the details provided are generally geared to large ADS development projects, a subset of these procedures can be tailored to the smallest of development efforts.

6-2. PURPOSE. To discuss guidelines for establishing and implementing the test program.

6-3. INFORMATION. The test program has two major phases; development testing and validation testing. Development testing includes module and integration testing. Validation testing is conducted by a test group independent of the developer during the test phase. A variation of validation testing has been successfully implemented on large multisite ADS development projects. This variation is described in paragraphs 6-6 and 6-7, and is composed of two parts. Part One is an Environmental System Test, Phase I (EST-I), conducted in as near a "real world" environment as possible Part Two is an Environmental System Test, Phase II (EST-II) conducted at one or more "live" operational sites by the users. Both EST-I and EST-II may be used to

provide validation testing functions and lead to an operational capability.

6-4. DEVELOPMENT TESTING.

a. GENERAL. Development testing is conducted by the programmers and analysts responsible for development of the CPCI. Testing is initiated when a particular module has been coded, compiled, and verified to the extent that the module accepts normal inputs and provides outputs. During development testing, emphasis is placed on testing discrete logic sequences; determining accuracy of computation; showing repeatability of result; testing upper, lower, and nominal ranges of data values; testing error conditions; verifying timing, sizing, interfaces, and data handling characteristics on a module-by-module basis; and integrating the modules and test the CPCI as a whole.

b. DEVELOPMENT TEST PLAN. All development test activities are conducted against development test plans. These plans are prepared by the development programmers and analysts during the development phase and form the basis for the test plan (PT), described in DODM 4120.17M. Normally, development test plans are prepared prior to the preliminary design review (PDR) and test procedures added prior to the critical design review (CDR). They should be reviewed by appropriate project members prior to the start of development testing.

(1) Development test plans (Attachment 2) are issued for each CPCI, in accordance with project schedules.

(2) The intent is to have a well coordinated test program to insure that complete and verified software is delivered to the independent test group at the completion of development testing. In many cases, the highest level of development tests are dry runs for the validation tests that follow. Development test plans should be coordinated within the development organization to insure these goals are achieved.

c. DEVELOPMENT TEST REPORTING. The project manager should be provided a report on: the number of tests executed, analyzed, and validated; tests with problems; tests to be rerun; assessment of problems on the schedule; etc. This reporting process is to gain insight as to the progress of testing amd to determine if the schedule is being met. Any change made to the software or to its documentation as a result of testing is noted and incorporated in an update of the program specification (PS) to be issued after development testing.

6-5. VALIDATION TESTING. This is the formal testing that occurs prior to delivery of the software. It is normally performed by a test group, independent of the developers. Emphasis is on validating that the requirements of the FD and the SS are met. The main purpose of this testing is to demonstrate that actual performance meets required performance in as nearly an operational environment as is practical. All validation testing should be conducted using operational system software and hardware. Sufficient resources and

time must be allocated to the independent test group.

a. TEST PLAN (PT). The PT defines the test program to be conducted by the independent test group. It specifies the purpose, objectives, and overall description of testing to be performed. Detailed test procedures are also included which govern the conduct of individual tests. The PT is prepared by the development activity and coordinated with the independent test group prior to initiation of the test phase. The independent test team may augment the PT with additional tests, as required. The PT is prepared in accordance with DODM 4120.17M.

b. TEST EXECUTION.

(1) Test Teams. The independent test group usually is organized into test teams, with each team being given the responsibility for validating a selected subset of the ADS. To insure the test teams become as familiar as possible with their subset, the assignment of test teams should be made early enough to allow each team to become thoroughly knowledgeable in the functions and operation of its subset of the software. Additional personnel should be added to the teams as the project matures. Each test team consists of members of the independent test group, with one designated as team leader. Generally, it is beneficial to augment the test teams with several developers and users who can support the analysis of test runs. However, responsibility for testing rests with the test team leader.

(2) Control of Test Execution.

(a) The test plan contains a master schedule of the planned tests. This schedule indicates the earliest planned execution date, and provides for a reasonable number of reruns of each case. The schedule is prepared on the basis of critical test dependencies, computer time requirements and availability.

(b) During testing, a record should be maintained of the number of times a test is executed and the reasons for reruns. The software configuration used for each run should be identified. Status of the analysis of each test is also recorded to give the test group, the development group, and the project manager visibility into the progress of testing.

(c) For large development projects, it is advantageous to establish a test review board (TRB) composed of developer and user representatives. This board performs the primary test monitoring function during formal testing, and expedites any corrective action required.

(d) Changes in the data base which are required to facilitate testing must be coordinated through the software developers, the project manager and, in some instances, the user. The data test package should contain a record of the data base configuration used for each test.

(e) The test procedures are the key documents to be controlled during the test phase. Approval authority for changes to test procedures depends on the extent of the change. Any change that relates to specification requirements must be approved by the configuration control board (CCB), while the test review board or test team leader may be the approval authority for all other changes that do not have a cost and schedule impact.

e. TEST REPORTING.

(1) A test status report should be prepared and distributed as frequently as the size of the project requires. The material in this report will be essentially the same as that presented in test review board (TRB) meetings, if the project has established such a board. For each reporting period, the test status report will identify:

(a) Tests executed.

(b) Tests analyzed and validated.

(c) Tests having open software problems.

(d) Tests to be rerun.

(e) Status of test reruns.

(f) Review of open software problems.

(g) Assessment of impact of open software problems.

(h) Forecast of tests to be executed during next reporting period.

(i) Summary of complete test schedule.

(2) A supplemental report which is essential during testing is the software problem summary. Each Software Problem Report (SPR) written during testing is identified by its control number, the module affected, the date the SPR was logged, and the date the SPR was closed.

(3) The final reporting instrument for each test procedure is Section 2 of the test analysis report (RT). When analysis has been completed, the final test procedure, test results and any change control documentation generated as a result of the test execution, will be provided to those personnel responsible for preparation of the final RT. At the conclusion of all testing, the RT is finalized and used as the primary input to the SVR.

6-6. ENVIRONMENTAL SYSTEM TEST, PHASE I (EST-I).

a. PURPOSE. The primary purpose of the EST-I is to insure the CPCI meets its specified requirements in a simulated "live" environment prior to operational release or further testing. EST I also allows an organization independent of the developer to perform the testing, check systems integration, and measure the impact on Data Processing Installation (DPI) operations. Successful completion of EST-I results in update of the product baseline and certification to release to EST II or for operational use.

b. PERSONNEL COMPOSITION/PARTICIPATION. The participants in EST I are quality assurance personnel, development activity representatives, and user representatives.

c. INPUTS/MATERIALS.

(1) Test Plan (PT).

(2) Users Manual (UM).

(3) Program Maintenance Manual (MM).

(4) Computer Operation Manual (OM).

(5) Computer programs and test data.

(6) Other documents/data, as required locally.

d. ACTIONS/DECISIONS.

(1) Examine documentation for compliance with standards.

(2) Perform tests in accordance with the PT.

(3) Analyze test results.

(4) Prepare a Test Analysis Report (RT).

(5) Assist with the retest of corrected errors.

e. OUTPUTS/MATERIALS.

(1) A Test Analysis Report (RT), in accordance with DODM 4120.17M.

(2) System for release to EST-II or for operational use.

(3) Updated Product Baseline.

(4) Other documents/decisions as required locally.

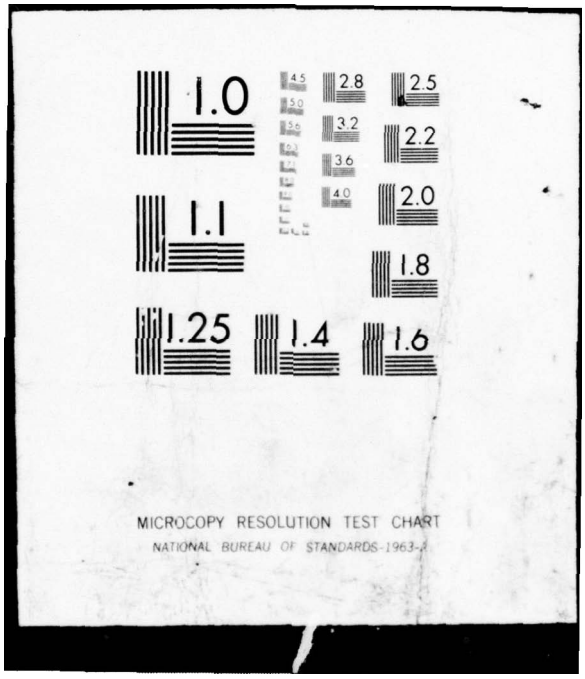
f. CONDUCT OF EST-I. The QA activity is responsible for the conduct of EST-I. The EST-I Test Analysis Report identifies problem areas which must be corrected prior to retesting. The developer and the user have the opportunity to review all test results to assure the products satisfactorily

AD-A052 126 DEFENSE SYSTEMS MANAGEMENT COLL FORT BELVOIR VA F/G 9/2
APPLYING PROGRAM MANAGEMENT CONCEPTS TO SOFTWARE DEVELOPMENT: A--ETC(U)
NOV 77 R CAUDILL

UNCLASSIFIED NL

2 OF 2
AD
A052 126

END
DATE
FILMED
5-78
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
8.0
9.0
10.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

meet the functional requirements. Corrections, if necessary, are made to the ADS programs and documentation, and the system undergoes additional testing, as required. When tests meet with the approval of all participants, the product baseline is updated.

6-7. ENVIRONMENTAL SYSTEM TEST, PHASE II (EST-II).

a. PURPOSE. The purpose of EST-II testing is to perform an evaluation of computer programs, documentation, and output products in a "live" environment at one or more selected operational sites. It allows base/MAJCOM data automation and functional personnel to perform their independent evaluations of the system with minimum assistance from the development activity.

b. PERSONNEL COMPOSITION/PARTICIPATION. The participants are quality assurance personnel, developer personnel, site personnel, user personnel, and when required, MAJCOM representatives.

c. INPUTS/MATERIALS.

(1) Test Plan (PT).

(2) Users Manual (UM).

(3) Program Maintenance Manual (MM).

(4) Computer Operation Manual (OM).

(5) System/Subsystem Release (Object Programs).

(6) Live-Test Data.

(7) Other documents/data as required locally.

d. ACTIONS/DECISIONS. The developer, in conjunction

with QA personnel and the users, nominates test sites. The test is conducted under the auspices of the QA activity with assistance of the developer, the base level functional and data automation personnel, and, when required, MAJCOM functional and data automation personnel. Problems are documented and corrections are validated by QA prior to release. When a successful System Validation Review (SVR) has been completed, the ADS is released.

e. OUTPUTS/MATERIALS.

(1) A Test Analysis Report (RT).

(2) Recommendation for conducting the SVR or further testing.

(3) Other documents/decisions as required locally.

f. CONDUCT OF EST-II. The developer, in conjunction with the QA activity and the user, nominates test sites. Coordination with MAJCOM and base level functional and data automation personnel is required to identify site adequacy to support the test requirements. The testing to be performed provides an evaluation of the capability of the ADS to operate in a "real world" environment and establishes the adequacy of the ADS to the user. The MAJCOM/ base level data automation and functional personnel will perform an independent evaluation of the ADS with minimum assistance from the developer. Problems will be documented and corrections validated by QA prior to release. When the test

activity is completed, the SVR will be conducted. Approval by the development activity and the user representative is required to successfully complete the SVR. Following the SVR, the ADS is released for operational use.

GLOSSARY

PART I - ABBREVIATIONS

ADS	Automated Data System
BCR	Baseline Change Request
CCB	Configuration Control Board
CDR	Critical Design Review
CI	Configuration Item
CM	Configuration Management
CMP	Configuration Management Plan
CPCI	Computer Program Configuration Item
DAR	Data Automation Requirement
DBCR	Data Base Change Request
DPD	Data Project Directive
DPI	Data Processing Installation
DPP	Data Project Plan
DPR	Design Problem Report
DS	Data Base Specification
DT	Development Test Plan
EST I	Environmental System Test, Phase I
EST II	Environmental System Test, Phase II
FCA	Functional Configuration Audit
FD	Functional Description
FOC	Full Operational Capability

HIPO	Hierarchy Input Process Output
MM	Program Maintenance Manual
OM	Computer Operation Manual
OS&M	Operational Support and Maintenance
PCA	Physical Configuration Audit
PDR	Preliminary Design Review
PS	Program Specification
PT	Test Plan
PVR	Product Verification Review
QA	Quality Assurance
RD	Data Requirements Document
RT	Test Analysis Report
SDR	System Design Review
SPR	Software Problem Report
SRR	System Requirements Review
SS	System/Subsystem Specifications
SVR	System Validation Review
TRB	Test Review Board
UM	Users Manual

PART II - DEFINITIONS

allocated baseline - The initial approved allocated configuration identification established at end of the definition phase.

baseline - A configuration identification document or set of such documents formally designated and fixed at a specific time during a CPCI's life cycle. Baselines, plus approved changes to those baselines constitute the current configuration identification.

baseline change request (BCR) - A request for alteration in the configuration of a computer program configuration item that is delivered or under development, after formal establishment of its configuration identification.

computer program configuration item (CPCI) - An ADS or portion of an ADS that is designated for configuration management.

configuration audit - a process to verify conformance to specifications and standards.

configuration control - The systematic evaluation, coordination, approval or disapproval, and implementation of approved changes in the configuration of a CPCI after formal establishment of its configuration identification.

configuration control board (CCB) - A board composed of representatives from program/project office and using/supporting organizations.

configuration identification - The currently approved technical description of the CPCI or ADS.

configuration item (CI) - An item of ADPE that is designated for configuration management.

configuration management (CM) - A management discipline that applies technical and administrative direction and surveillance to:

(a) identify and document the functional and physical characteristics of a configuration item.

(b) control changes to those characteristics.

(c) record and report configuration status.

configuration management plan (CMP) - A document which describes project responsibilities and procedures for implementing CM.

configuration status accounting - The recording and reporting of the approved configuration identification, the status of the proposed changes to the approved configuration, and the implementation status of approved changes.

critical design review (CDR) - A formal review conducted during the development phase before translating logic, and algorithms to coded instructions.

data base change request (DBCR) - A form used to initiate and control data base changes after the data base is placed under configuration control.

design problem report (DPR) - A form used for documenting problems identified during reviews and audits.

development testing - Testing of computer programs by the development programmers and analysts prior to EST I.

development test plan (DT) - A document which specifies the method and content for development testing from the lowest compilable level up through the complete computer program configuration item. Defines test management, reports, controls, manpower, acceptance criteria, and test procedures.

deviation A written authorization, granted prior to the development of a CPCI, to depart from a particular performance or design requirement; a specification for a specific number of units; a specific period of time; or established standards.

functional baseline - The initial approved functional configuration identification.

functional configuration audit (FCA) - The formal examination of CPCI to verify that the performance specified in the SS has been achieved.

module - A program unit that is discrete and identifiable with respect to compiling and combining with other units.

physical configuration audit (PCA) - The formal examination of the coded version of a computer program configuration item against its technical documentation.

preliminary design review (PDR) A formal review of the subsystem design approach for a CPCI occuring between the SDR and CDR.

product baseline - The initial approved product configuration identification.

product verification review (PVR) - A formal review conducted by the developer for each CPCI at the end of the development phase to establish the Product Baseline for that CPCI and to insure preparation for the Test Phase has been completed.

quality assurance (QA) - All actions that are taken to assure that a development organization delivers products that meet performance requirements and adhere to standards and procedures.

software problem report (SPR) - A form used to report a suspected or existing discrepancy or deficiency in an existing computer program, its operational documentation or interfacing hardware.

specification - A document that describes the requirements for the development or acquisition of ADPE and/or software.

system design review (SDR) - A formal review of the system design approach for an ADS.

system requirements review (SRR) - A formal review of the requirements for an ADS.

system validation review (SVR) - A formal review of the results of the Test Phase to insure that the ADS satisfies the requirements of the SS and FD.

test analysis report (RT) - A document containing the results and analyses of tests executed during the Test Phase.

waiver - A written authorization to accept a configuration item or other designated item that has been found to depart from specified requirements, but nevertheless is considered suitable for use as is or after rework by an approved method.

DEVELOPMENT TEST PLAN

The purpose of the development test plan is to describe the testing which is to be done during integration testing.

Chapter 1 - GENERAL INFORMATION

1-1. Summary. Summarize the purpose, scope, and intended use of the development test plan.

1-2. References. List applicable project documentation.

Chapter 2 - TEST MANAGEMENT

2-1. Roles and Responsibilities. Describe the roles and responsibilities of test participants in the development organization during the testing phase.

2-2. Reporting Procedures. Describe test reporting procedures to be employed.

Chapter 3 - TEST CONTROL.

3-1. Procedures. Establish test control procedures. Address:

a. The control and accounting system for test procedures, test data, and test execution.

b. Procedures for increasing control over software products as integration testing progresses.

c. Procedures for retest.

Chapter 4 - TEST DEFINITION

This section defines each test, acceptance criteria for the test, and procedures for conducting the test. The format

for presentation of test definitions is as follows:

4-1. Test Description. Describe the purpose of the test, required inputs/outputs, and acceptance criteria for measuring success of the test.

4-2. Test Identification. Assign a unique identifier and title for each test.

4-3. Environment. Describe the hardware and other software required to execute the test.

4-4. Test Procedures. Describe the procedures for conducting the test.

Chapter 5 - RESOURCE REQUIREMENTS

Discuss the equipment, personnel, and facilities that will be required to perform integration testing.

Chapter 6 - SCHEDULES

Provide a detailed schedule for development testing. The schedule should identify when individual modules are integrated into the CPCI.

PROGRAM MAINTENANCE MANUAL

PURPOSE: The primary purpose of program maintenance manual is to facilitate transfer of programs between individuals at any point in the life cycle of the programs. A secondary purpose is to rapidly determine program status at any point during development. The following guidelines are provided:

a. A program maintenance manual (MM) should be kept for every module.

b. The MM is prepared and maintained by the programmer or programmer team.

c. Specific format for the MM should be provided by the development organization using DOD Manual 4120.17M as a guide.

d. General guidelines for content include:

(1) Requirements Section. This section should include extracts from the FD and SS pertinent to the module. Additionally, a narrative should be included which explains how the module satisfies the requirements.

(2) Constraints Section. This section should include all known constraints, e.g., required core storage, peripherals required, table lengths, etc.

(3) Interface Section. Both internal and external interfaces should be included in this section. Extracts from the RD and DS should be used, if available.

(4) Design Section. Include logic or structure charts which show the complete logic flow of the module as well as all interface elements. The PS should be prepared from these charts if required by project management. A current program listing should also be included in this section.

(5) Test Section. This section includes test instructions and a description and the location of all items necessary for the conduct of module testing.

CONFIGURATION MANAGEMENT PLAN FORMAT

Section I - INTRODUCTION.

State the purpose of the configuration management plan (CMP) and briefly describe its contents.

Section II - PROJECT ORGANIZATION.

Describe the project organization and its relation to command structure. Identify the organizational level at which the configuration management function will be performed. A checklist for the preparation of this section is as follows:

a. Is configuration management at a suitable level in the project structure?

b. Are responsibilities clearly defined?

c. Are sufficient resources identified to perform the required tasks?

d. Is there adequate configuration management representation on the CCB?

Section III - CONFIGURATION IDENTIFICATION.

Identify responsibilities for establishing configuration identification. Identify the CPCI(s) to be controlled and related documentation. Identify responsibilities for preparing the documentation plan. A checklist for the preparation of this section is as follows:

a. Does the CMP identify the documents that establish the baselines?

b. Are the responsibilities for preparing and approving project documentation clearly defined?

c. Does the documentation plan indicate when project documentation is placed under configuration control?

d. Does the schedule in the documentation plan meet project and user requirements?

e. Does the CMP include an appropriate numbering scheme for the software products produced?

Section IV - CONFIGURATION CONTROL.

Define procedures for control of CPCIs and for processing changes thereto. Include change control procedures for processing changes generated by the development activity and other organizations. A checklist for the preparation of this section is as follows:

a. Does the CMP include an adequate change control system for the various classes of changes established?

b. Is a method provided for controlling the interfaces between elements of the CPCI during development and testing?

c. Does the CMP identify methods for controlling interfaces with external ADS?

d. Is the change control process clearly described with flow charts and sample forms?

e. Is there an adequate procedure described for storage and physical control of software products?

f. Is there an adequate procedure described for release of software products into a controlled environment?

Section V - CONFIGURATION STATUS ACCOUNTING.

Describe the recording and reporting procedures and forms to be used for configuration status accounting. Define any automated methods to be used in status reporting. A checklist for the preparation of this section is as follows:

a. Does the CMP identify the data required for configuration status accounting, and the method for recording it?

b. Does it identify the status accounting reports required, and their frequency and distribution?

c. Are forms, formats, and reports clearly identified and included?

Section VI - REVIEWS AND AUDITS.

Describe plans for configuration management support to reviews and audits. Define products and documents to be reviewed or audited, reviewing authority, method for handling deviations or waivers, change procedures and forms, and numbering changes.

Section VII - OPERATIONAL IMPLEMENTATION.

Define procedures for maintaining the product baseline after initiation of the operation phase.

TYPES OF CONFIGURATION STATUS

ACCOUNTING LOGS

1. PRODUCT LOG: The product log is used to list and describe existing tapes, decks, and listings of released software:

a. Tape identification.

b. Content of tape, e.g., module IDs, CPCI identification, etc.

c. Listings and card decks controlled and numbered by the module ID.

d. Originator name and date. Tape library number (if applicable).

2. MODULE LOG. The module log is a history of each module's development.

a. Module ID.

b. List the BCR/SPR/DBCR that resulted in changes to the module.

c. Tape number and tape library number, if applicable.

d. Program specification (PS), title and number, in which the module is described.

3. SOFTWARE PROBLEM REPORT (SPR) LOG. The SPR log is used to list and describe all SPR data, including fixes and other closing information:

a. SPR number and date.

b. SPR originator's name.

c. Module ID and and data base information.

d. Organization responsible for action.

e. Status.

4. DATA BASE CHANGE REQUEST (DBCR) LOG. The DBCR log is used to list and describe all DBCRs:

a. DBCR number and date.

b. DBCR originator's name.

c. Data base identification.

d. Identification and date of data base that incorporates the change.

5. BASELINE CHANGE REQUEST (BCR) LOG. The BCR log is used to list BCRs.

a. BCR number and date.

b. Originator's name.

c. BCR classification (Class I or II).

d. Baseline on which BCR was originated.

e. Baseline documentation affected.

f. Approval/disapproval and date.

g. Date implemented, if approved.

6. DESIGN PROBLEM REPORT (DPR) LOG. The DPR log is used to list all DPRs submitted during reviews and audits:

a. DPR number and date.

b. Originator.

c. Organization responsible for action.

d. Date DPR closed.

CONFIGURATION CONTROL FORMS

PURPOSE: To provide instructions for completion of configuration control forms.

a. DESIGN PROBLEM REPORT (DPR)

CONTROL NUMBER: Enter a unique control number in accordance with local configuration management procedures.

DATE: Enter date submitted.

ORIGINATOR: Individual who originated the DPR, his organization, and phone number.

DOCUMENT: Identify the document in which the problem exists.

PAGE/SECTION/FIGURE: Identify the specific reference in the document cited above.

PROBLEM/RECOMMENDATION: Describe the problem and recommendation resulting from the review in sufficient detail for corrective action to be taken.

ACTION CATEGORY: Reviewing development organization check one of four indicated categories.

ACTION ITEM NUMBER: Number assigned in accordance with local procedures as a cross reference to the minutes of the review.

DISPOSITION: Indicate action taken or to be taken to resolve the DPR.

CONTROL NO. _____

DESIGN PROBLEM REPORT (DPR) DATE _____

Originator: _____

Document _____

Page/Section/Figure _____

Problem/Recommendation:

Action Category

Design 1. ☐

Incorrect 2. ☐

Incomplete 3. ☐

Editorial 4. ☐

Action Item No.

Disposition:

Closed ☐

b. SOFTWARE PROBLEM REPORT (SPR)

CONTROL NUMBER: Enter a unique control number in accordance with local configuration management procedures.

DATE: Enter date SPR submitted.

TO: Enter the organization responsible for development or maintenance of the software.

FROM: Enter the initiating organization.

INFO COPIES TO: Self-explanatory.

PROGRAM NAME: Enter the name of the program in which the problem or discrepancy was detected.

IDENT: Enter the identification of the program involved.

ADS: Enter the title of the ADS of which the program is a part.

RUN DATE: Enter the date the program was run in which the discrepancy or error was detected.

POINT OF CONTACT: Enter the name of the individual in the organization which initiated the SPR who is most familiar with the problem.

PROBLEM DESCRIPTION: Describe what the discrepancy or error is and circumstances which contributed to the cause. Describe what should have occurred had there been no discrepancy and the impact of the discrepancy.

COMMENTS: Indicate the urgency of the correction and any other pertinent facts.

PROBLEM ANALYSIS: Describe the cause of the discrepancy or error, the impact, and any other programs or data bases affected.

RECOMMENDED ACTION: Describe the proposed corrective action (if necessary), and provide an estimate of the time and resources required to complete the recommended action.

APPROVED: Self-explanatory.

DISAPPROVED: Self-explanatory.

SIGNATURE: Self-explanatory.

DATE: Enter date of signature.

ACTION TAKEN: Indicate what corrective action was taken if such action was necessary. Additionally, include the date the action was completed.

Control Number _____

SOFTWARE PROBLEM REPORT Date _____

TO _____

FROM _____

INFO COPIES TO _____

Program Name _____ IDENT _____ ADS _____

Run Date _____ Point of Contact _____

PROBLEM DESCRIPTION

COMMENTS

Software Problem Report

Problem Analysis Analyst _____ Date _____

Recommended Action

☐ Approved

☐ Disapproved Signature _____ Date _____

Action Taken

c. DATA BASE CHANGE REQUEST (DBCR)

CONTROL NUMBER: Enter a unique control number in accordance with local configuration management procedures.

DATE: Enter the date the DBCR is submitted.

TO: Enter the organization responsible for accomplishing the requested change.

FROM: Enter initiating organization and the name and phone number of the individual in the organization initiating the DBCR who is most familiar with the request.

DATA BASE IDENTIFICATION: Clearly identify the data base to be changed through implementation of the request.

MODULE(S) AFFECTED: Identify all modules affected to insure all necessary action is taken.

FORM OF INPUT: Check one listed. If "other," describe in REMARKS block.

REASON FOR CHANGE: Self-explanatory.

REMARKS: Describe form of input if "other" is checked above; any additional information which could assist in accomplishing the change.

CHANGE IMPLEMENTED: This block is to be completed by the appropriate review official when change is accomplished.

DATA BASE CHANGE REQUEST

TO:

Control Number _____
Date of Request _____

FROM:
Organization _____
Point of Contact _____ Phone _____

DATA BASE IDENTIFICATION

MODULE(S) AFFECTED _____

FORM OF INPUT

- ☐ D/B Input Requirements Form
- ☐ Data Cards and Listings
- ☐ 80-column coding sheets
- ☐ D/B Tape Containing Data Block, Reel Number _____
- ☐ Other (Specify Below)

REASON FOR CHANGE

REMARKS

CHANGE IMPLEMENTATION

☐ APPROVED SIGNATURE _____ DATE _____

d. BASELINE CHANGE REQUEST (BCR)

PROJECT TITLE: Enter title of project.

CHANGE CONTROL NUMBER: Enter a unique control number in accordance with local configuration management procedures.

DATE: Enter date submitted.

OFFICE SYMBOL: Enter office symbol of originator.

TELEPHONE NUMBER: Enter telephone number of change request originator.

CHANGE CLASS: Self-explanatory.

BASELINE AND BASELINE DOCUMENTS REQUIRING CHANGE: Check the baseline(s) or document(s) affected by the change request.

DESCRIPTION OF CHANGE: A description of the requested change will be sufficient in detail to allow for determination of project impact. Other pertinent facts may be attached.

JUSTIFICATION: This will contain information sufficient to support the requested change. Generalities must be avoided.

OBJECTIVES/BENEFITS: Explain what should be achieved by this change. Quantify the results of this change in terms of money, time, equipment, etc.

PROJECT IMPACT: Enter such information as increased costs, schedule delays, product changes, etc.

COST ESTIMATES CHANGE: Provide estimates in terms of manhours the change will require. Only those costs incurred by the change will be shown and not total development costs. However, the total costs prior to this change may be shown for comparison purposes.

DEVELOPER SIGNATURE: The appropriate approval/disapproval block will be checked, developer (usually project officer) will sign, insert his office symbol and date of signature.

USER SIGNATURE: The appropriate approval/disapproval block will be checked, user will sign, insert office symbol and date of signature. It is not necessary to obtain user signature for Class II changes.

MANAGEMENT REVIEW: Enter the review that evaluated this change (CCB or SDR). The approval official enters signature, checks either the approved of disapproved block, and enters the date of approval or disapproval.

BASELINE CHANGE REQUEST

PROJECT TITLE	CHANGE CONTROL NUMBER	DATE

ORIGINATOR	OFFICE SYMBOL	TELEPHONE

CHANGE CLASS

I ☐ II ☐

BASELINES AND BASELINE DOCUMENTS REQUIRING CHANGE

☐ FUNCTIONAL ☐ ALLOCATED ☐ FD ☐ SS ☐ RD ☐ OTHER

☐ PRODUCT ☐ PS ☐ UM ☐ OM

DESCRIPTION OF CHANGE

JUSTIFICATION

BASELINE CHANGE REQUEST

OBJECTIVES/BENEFITS

PROJECT IMPACT STATEMENT

COST ESTIMATES

- FUNCTIONAL ANALYST MANHOURS
- DATA SYSTEM ANALYST MANHOURS
- PROGRAMMER MANHOURS
- OTHER MANHOURS

TOTAL MANHOURS

COMPUTER SUPPORT HOURS

OTHER COSTS

DEVELOPER SIGNATURE	☐ APPROVED	☐ DISAPPROVED	OFFICE SYMBOL	DATE
USER SIGNATURE	☐ APPROVED	☐ DISAPPROVED	OFFICE SYMBOL	DATE
MANAGEMENT REVIEW	☐ APPROVED ☐ DISAPPROVED	SIGNATURE		DATE

BIBLIOGRAPHY

Articles

1. Boehme, Barry, PhD, "Software and Its Impact: A Quantitative Assessment", Datamation, May 1973

2. Chen, Gordon K.C., "What is the Systems Approach?", Selected Computer Articles, DODCI, July 1976

3. Gunderson, James R., PhD, and McMurray, Frank W., "Making Project Management Effective", Selected Computer Articles, DODCI, June 1976

4. Keider, Stephen P., "Why Projects Fail", Selected Computer Articles, DODCI, July 1976

5. McCarthy, Rita, "Applying the Technique of Configuration Management to Software", Defense Management Journal, October 1975

6. Miller, Donald R., LtCol, USMC, "Information Cost Awareness", Selected Computer Articles, DODCI, July 1977

7. Mintzberg, Henry, "The Manager's Job: Folklore and Fact, Selected Computer Articles, DODCI, July 1977

8. Nolan, Richard, L., "Business Needs a New Breed of EDP Manager", Selected Computer Articles, DODCI, July 1977

9. Rodgers, Tommy R., "Management Involvement Pays Off", Selected Computer Articles, DODCI, July 1976

10. Sprague, Ralph H., PhD and Watson, Hugh T., PhD, "Misconcepts", Selected Computer Articles, DODCI, July 1976

Speeches

11. Carabello, John M., Director for Data Automation, OASD, Speech entitled Life Cycle Management, delivered July 1977

12. Fife, Dennis W., "Speech to Seminar of Computer Productivity - July 1977", Employed By, Institute for Computer Sciences and Technology; National Bureau of Standards, Washington, DC

13. Martins, Cecil, Major, USAF, Gunter AFS AL, Briefing prepared to capsulize PhD dissertation research, summer 1976