

**COMMON USER BASELINE FOR THE
INTELLIGENCE COMMUNITY (CUBIC)**

**CONFIGURATION MANAGEMENT PLAN
VERSION 3.0**

9 August 2001

Produced By:
Department of the Air Force
Air Force Research Lab
Rome Research Site
32 Brooks Road
Rome, New York 13441-4114

TABLE OF CONTENTS

1.	INTRODUCTION	1
1.1	BACKGROUND.....	1
1.2	PROGRAM OVERVIEW	2
1.3	DOCUMENT OVERVIEW	2
1.4	DOCUMENTATION UPDATES	3
2.	REFERENCE DOCUMENTS	4
3.	TERMS.....	5
4.	CONCEPT OF OPERATIONS FOR CUBIC CONFIGURATION MANAGEMENT	8
4.1	THE CUBIC CM PROCESS	8
5.	CONFIGURATION CONTROL MANAGEMENT STRUCTURE	11
5.1	PROGRAM MANAGEMENT OFFICE.....	11
5.2	PROGRAM CONFIGURATION CONTROL BOARD (CCB).....	11
5.2.1	Program CCB Responsibilities.....	12
5.2.2	Program CCB Attendees	12
5.2.3	Program CCB Agenda.....	12
5.2.4	Program CCB Minutes	13
5.3	CUBIC CONFIGURATION CONTROL BOARD	13
5.3.1	CCCB Responsibilities.....	13
5.3.2	CCCB Attendees	13
5.3.3	CCCB Agenda.....	13
5.3.4	CCCB Minutes	14
6.	CONFIGURATION MANAGEMENT ACTIVITIES.....	15
6.1	PRODUCT BASELINE VERSION IDENTIFICATION	15
6.2	DOCUMENT/MEDIA IDENTIFICATION.....	15
6.3	CONFIGURATION CONTROL	16
6.3.1	Flow of Configuration Control.....	16

6.3.2	Configuration Control Documentation.....	17
6.3.3	Documentation Storage and Control	18
6.3.4	Document Reproduction Requests	18
6.3.5	Software Storage and Control	18
6.3.6	Software Reproduction Requests	18
6.4	CONFIGURATION STATUS ACCOUNTING.....	19
6.4.1	CMDB	19
6.4.2	CMDB Access.....	20
6.5	METRICS.....	20
6.6	QUALITY ASSURANCE.....	20
6.7	REVIEWS AND AUDITS	21
6.7.1	DODIIS Certification Process	21
6.8	CONFIGURATION INSTALLATION AND TESTING	21
7.	TEST FINDING (TF) PROCESS	23
7.1	IMPACT CODE DESCRIPTIONS.....	23
7.1.1	JITF	23
7.1.2	Security.....	25
8.	PROBLEM REPORT/CHANGE REQUEST/MULTIPLE APPLICATION PROBLEM/REQUIREMENT (PCMR) PROCESS	26
8.1	CM RESPONSIBILITIES	26
8.2	PCMR PROCESS DESCRIPTION.....	26
8.2.1	Impact Code Descriptions	30
8.2.2	Classification Issues	30
8.2.3	Submission of an Impact Code 1 PR.....	31
8.2.4	PCMR Status Values.....	32
9.	DOCUMENT MANAGEMENT PROCESS	34
9.1	DRR PROCESS DESCRIPTION	34
9.1.1	DRR Status Values.....	35
9.2	CDRL TRACKING PROCEDURES.....	36
10.	ACTION ITEM PROCESS.....	37
10.1	AI PROCESS DESCRIPTION.....	37

10.1.1 Action Item Status Values	38
11. SOFTWARE RELEASES	39
11.1 PMO RESPONSIBILITIES	39
11.2 SITE RESPONSIBILITIES	39
11.3 CM RESPONSIBILITIES	40
11.4 DEVELOPMENT CONTRACTOR RESPONSIBILITIES	40
11.5 EMERGENCY RELEASES	40
12. ACRONYMS	41

TABLE OF FIGURES

Figure 4-1 CUBIC CM Process	9
Figure 5-1 Configuration Control Management Structure and Information Flow	11
Figure 6-1 Document Media Number	16
Figure 6-2 DODIIS Certification Process	22
Figure 7-1 Test Finding Process	24
Figure 8-1 PR/CR Process	28
Figure 8-2 Impact Code 1 PR Process	31
Figure 8-3 PR/CR Status Values	33
Figure 9-1 DRR Process	35
Figure 9-2 DRR Status Values	36
Figure 10-1 Action Item Process	37
Figure 10-2 Action Item Status Values	38

1. INTRODUCTION

This document provides Configuration Management (CM) direction and applies CM discipline to the development, installation, maintenance, modification, and enhancement of software and documentation for multiple programs. The objective of this CM Plan is to describe a process for more effectively developing and maintaining software by improving accountability, reproducibility, traceability, and coordination.

The services described in this document can be tailored to meet the needs of an individual program. In addition, Program Management Offices (PMOs) can “pick and choose” which CM services to receive and have maintained. All information contained in this document is superseded by information contained in the program contract regarding the same subject areas. Program managers are encouraged to incorporate the concepts promoted in this document into their software development contracts.

This plan establishes CM policies, methods, and procedures to be implemented and followed to control the configuration of software programs managed by Air Force Research Laboratory (AFRL), Electronic Systems Center (ESC), and Wright Patterson Air Force Base. This document supersedes the *Common User Baseline for the Intelligence Community (CUBIC) Configuration Management Plan, Version 2.0*, dated 5 November 1999.

Additional information on the items discussed in this document can be obtained by contacting CM at the following address and phone number:

Configuration Management
AFRL/IFEB
32 Brooks Road
Rome, NY 13441-4114

DSN 587-2723/4209
COMM 315-330-2723/4209
FAX 315-330-1637

UNCLASS E-MAIL cubic_cm@rl.af.mil
CLASS E-MAIL cubic.cm@mail.rome.ic.gov

INTERNET www.if.af.mil/programs/cm/
INTELINK web1.rome.ic.gov/cm/

1.1 BACKGROUND

The AFRL, Rome Research Site, is home to a number of Air Force and Joint PMOs responsible for developing Department of Defense (DOD) intelligence software and the Joint Integration Test Facility (JITF). AFRL's Information Handling Branch (IFEB), working with the Aerospace Command and Control Intelligence Surveillance and Reconnaissance Center (AC2ISRC/A-2), formerly 497IOG/IND, provides a unique collection of CM services supporting these PMOs through the CUBIC organization.

CUBIC has a strong legacy of supporting the intelligence community through defined processes for problem identification, change control, and quality software distribution. The success of CUBIC is predicated upon extensive user participation throughout the software development cycle. These concepts have evolved since their initial inception to a streamlined, focused, cost-effective methodology that assists the PMO in controlling changes to software baselines and provides critical information to multiple users.

1.2 PROGRAM OVERVIEW

The foundation of CUBIC is the successful partnership between AC2ISRC/A-2, formerly 497IOG/IND, and AFRL. The AC2ISRC/A-2 functions as CM's executive manager and ensures representation of user requirements at upper levels of DOD management. AFRL/IFEB provides CM services to PMOs and their users and supports the AC2ISRC/A-2. This process supports PMOs at AFRL, ESC at Hanscom Air Force Base, Wright Patterson AFB, and users at over 200 locations throughout the world. Roles and responsibilities for each of these organizations and CM participants are outlined in this document. The services provided include:

- Receiving, processing, and accounting for the status of Problem Reports and Change Requests (PRs/CRs), Multiple Application Problems (MAPs), Requirements, Document Review Reports (DRRs), Test Findings (TFs), and Action Items (AIs).
- Identifying, controlling, and accounting for the status of software configurations.
- Maintaining software and documentation libraries for supported programs.
- Supporting programs with software release and documentation distribution.
- Maintaining application baselines.
- Streamlining and standardizing PMO/user communications by implementing uniform reporting and tracking procedures and practices, including on-line global network data files and interfaces.
- Maintaining and enhancing interoperability between software applications through management oversight at the community level.
- Using and expanding automated and networked management aids to streamline CUBIC processes and interactions with worldwide users/developers.

The CUBIC process acknowledged more than a decade ago that "stovepipe" development was counter-productive to the success of intelligence community goals. For this select subset of intelligence applications there is guidance for the implementation of major DOD initiatives and downward directed requirements through formal change control procedures. There is also a primary focus on upward directed requirements identified by operational users. The CM process begins and ends with the user and is based upon extensive user involvement. CUBIC provides its users with the mechanisms and support to enhance the performance of the software applications required by the intelligence mission.

1.3 DOCUMENT OVERVIEW

The contents of this document are organized as follows:

Section One: Scope of Configuration Management provides an overview of CM, what purpose it serves, related terms, and highlights the structure of CM.

Section Two: Referenced Documentation provides a listing of documentation supporting CM processes or referenced by the CM Plan.

Section Three: Terms of Reference provides definitions of terms used throughout the CM Plan.

Section Four: CUBIC Concept of Operations provides an overview of how all the CM processes work together to provide quality software releases.

Section Five: Configuration Control Structure describes the change management structure and the flow of information between organizations managing supported programs.

Section Six: Configuration Management Activities describes the support function of CM activities and the tool used in tracking necessary program life cycle information.

Section Seven: Test Finding Process describes the identification, tracking and management process for TFs generated during formal integration, interoperability, and security testing.

Section Eight: Problem Report/Change Request/Multiple Application Problem/Requirement Process describes the process in which PRs, CRs, MAPs, and Requirements are generated and tracked. This is a planned and systematic set of techniques, documentation, and actions necessary to control change to individual and related applications.

Section Nine: Document Management Process describes the use of DRRs in evaluating and verifying the accuracy and adherence to standards for documentation releases. Contract Data Requirements List (CDRL) tracking procedures, which help monitor products delivered in accordance with contract requirements, are also described.

Section Ten: Action Item Process describes the use of AIs as a management tool to identify and track issues for supported programs.

Section Eleven: Software Releases describes the software release process.

Section Twelve Acronyms provides definitions for the acronyms used in this document.

1.4 DOCUMENTATION UPDATES

This CM Plan is a living document in that the processes described are continuously evaluated to determine if enhancements are possible. Changes to the policies described in this document are implemented after coordination with the CM program manager, CM executive manager, and the Chief of the Information Handling Branch. The Branch Chief oversees a significant proportion of the programs currently supported by CM, and is, therefore, a part of the process. PMOs and sites are encouraged to submit their suggestions for enhancements, new services, or other changes to the CM office for consideration using the DRR process, described in Section Nine.

2. REFERENCE DOCUMENTS

The following documents are referenced or are used in support of this CM Plan:

- CJCSI 6212.01A, Compatibility, Interoperability, and Integration of Command, Control, Communications, Computers, and Intelligence Systems, 30 June 1995.
- Defense Intelligence Agency Regulation No. 65-13, Automated Information System Life Cycle Management.
- DOD 5000.2, Operation of the Defense Acquisition System, 23 October 2000.
- DOD Directive 5000.1, Defense Acquisition, 23 October 2000.
- Department of Defense Intelligence Information System (DODIIS) Instructions, February 2000.
- IEEE/EIA 12207.0-1996, Industry Implementation of International Standard ISO/IEC 12207: 1995, March 1998.
- IEEE/EIA 12207.1-1997, Industry Implementation of International Standard ISO/IEC 12207: 1995, April 1998.
- IEEE/EIA 12207.2-1997, Industry Implementation of International Standard ISO/IEC 12207: 1995, April 1998.
- Joint DODIIS/Cryptologic SCI Information Systems Security Standards, 28 March 1997.
- Director of Central Intelligence Directive 6/3 Protecting Sensitive Compartmented Information within Information Systems
- Test and Evaluation Policy for DODIIS Intelligence Mission Applications (IMAs), April 1999.
- User's Manual for the Configuration Management Database (CMDB) Version 2.0, 01 March 1999. (new date)
- EIA-649 Standard - National Consensus Standard for Configuration Management, August 1998

3. TERMS

The following is a list of definitions for terms used in this document.

Action Item (AI) - A management tool used to identify and track program issues.

Address Indicator Group (AIG) - A collection of Automatic Digital Network (AUTODIN) message Plain Language Addresses (PLAs) for a related group of users. The relationship can be a common software application, mission, or organizational alignment (such as all DODIIS users). An AIG facilitates a quick and efficient means of communication between the PMO and users, keeping everyone cognizant of issues that concern a program. AUTODIN is in the process of being replaced by the Defense Messaging System (DMS).

Change Request (CR) - A new requirement or enhancement identified for addition to the baselined software requirements.

Compatibility - The capability of two or more items or components of equipment or material to exist or function in the same system or environment without mutual interference.

Configuration Management Database (CMDB) - An automated status accounting tool used to record and report information related to supported programs by CM.

Development Contractor - The organization or company developing software under contract with the Government.

Document Review Report (DRR) - A tool used to identify discrepancies and manage change to documentation.

DODIIS Executive Agents (DEXAs) - Office [Service, Agency, or Unified & Specified (U&S) Command] responsible for management and requirements oversight of one or more software programs.

Executive Manager - Office responsible for requirements oversight of one or more programs. If there is a DEXA for the program, he/she is the executive manager. Program requirements are verified and validated by the executive manager.

Integration - The arrangement of applications in an architecture so that they function together in an efficient and logical way.

Intelligence Mission Application (IMA) - An automated information system within a corporate information management functional area selected as the standard application to support processes for a functional activity. IMAs were formerly known as migration systems and are also identified as applications, information technology components, and/or tools.

Interface - The functional and physical characteristics required at a common boundary between two or more hardware/software products.

Interoperability - The ability of the systems, units, or forces to provide services to and accept services from other systems, units, or forces, and to use the services so exchanged to enable them to operate effectively together. The conditions achieved among communications-electronics

systems or items of communications-electronics equipment when information or services can be exchanged directly and satisfactorily between them and/or their users.

Legacy System - An application whose critical functionality will be subsumed by an IMA or an application that duplicates the functions of an IMA and is scheduled to be terminated, e.g., DAWS and MAXI.

Library - A centralized collection of documentation and software for supported programs. Documents are available hardcopy, softcopy, or on-line.

Multiple Application Problem (MAP) - The mechanism used to coordinate changes between applications when a PR or CR affects multiple applications.

Notification Letter - A formal request from the PMO to the contracted developer requesting a written plan on the level of effort to implement a change to the software.

Plain Language Address (PLA) - An address consisting of the site name, its location, and any appropriate office symbols, e.g., AFRL ROME NY//IFEB//. Each office symbol listed on an AUTODIN message PLA will receive a copy of all messages addressed to the AIG.

Point of Contact (POC) - An individual with interest in or responsibility for one or more supported programs. POC information is maintained by CM to define a user, site, organization, PMO, or developer.

Problem Report (PR) - A report describing a software deficiency where the software does not function as documented by the program requirements.

Profile - A Profile is the information maintained on individuals provided access to the CMDB. The profile contains information on POCs, street and electronic mailing addresses, phone numbers, AUTODIN message PLAs, facsimile numbers, and programs of interest.

Program - All the activities and processes involved in creating a software application.

Program Management Office (PMO) - The PMO is responsible for developing an acquisition strategy, planning the program by developing a management approach, budgetary estimates and alternatives, program schedules, contract strategies and structures, and conducting the day-to-day management of the program's development, enhancement, or maintenance. The PMO is also responsible for directing the development contractor, testing and fielding of the application.

Program User Group - A group to provide a forum for information exchange between the users and program management. The users can make recommendations to the PMO as appropriate. During user group meetings, representatives from program management, users, and contractor(s) meet to discuss program specific issues. These issues may include, but are not limited to program status, proposed enhancements, problems and concerns, CM, user issues, and demonstrations. Program user group meetings are normally held semi-annually and can be hosted by the PMO or at a designated location.

Requirement - Information maintained by CM to track program level requirements for future versions. These requirements translate to program specifications.

Site - The physical location where software applications are installed and operated by users.

Software Problem Report (SPR) - A report describing a software deficiency where the software does not function as documented by program requirements during design reviews and development contractor-run testing.

Software Release - A software release is made up of the software application media and associated documentation. CM tracks and stores detailed information regarding software releases and supports reproduction and distribution of releases for programs.

Software Version Description (SVD) - The document that identifies the exact version and contents of the software release packages and contains the following information: document identification, inventory and description of release package contents, PR/CR change summary, and interface compatibility.

Tasking Letter - Formal tasking from the PMO to the development contractor to implement a software change into a specified version of software.

Test Finding (TF) - A software or document deficiency identified during formal testing, i.e., JITF, Joint Interoperability Test Command (JITC), or Security, of software applications.

User - Any organization or individual that operates or is affected by software applications supported by CM.

Vendor - An organization or company that supplies Commercial-Off-The-Shelf (COTS) and/or Government-Off-The-Shelf (GOTS) software to the government.

Workplan - A document that identifies the estimated level of effort required to implement a software change.

4. CONCEPT OF OPERATIONS FOR CUBIC CONFIGURATION MANAGEMENT

CUBIC CM is designed to facilitate the decision making process and information flow between PMs responsible for developing software and the users of that software. The CUBIC CM team functions as a “force multiplier” by enhancing activities of fifteen supported PMOs and providing an economy of scale. CM deals with CM issues everyday while most PMOs focus on documentation or problem reporting only during specific points in their program’s lifecycle.

CM support is fundamentally the same regardless of the scope of change (i.e. one PR fix versus a major release of software), the development methodology (i.e. spiral versus “traditional”), or the size of the program. CUBIC CM provides a standard to which supported programs are developed and supported throughout their lifecycle. All CM activities; processing PR/CRs, DRRs, Action Items, tracking Software Release and POC information, documentation management, and distribution are designed to ensure that only quality software is released to the field. CM ensures clear tracking for which changes are made to software and why, what related items are changed as a result of the software change, and what the level of effort is to change the software. CM also tracks where software has been sent, what versions are running in the field, and who is responsible for the software at site.

Most importantly, CM provides a history of what has been done for a program. CM identifies which methods worked, and which caused failures so that past successes can be built upon and past mistakes can be avoided. CM also provides methods for planning for the future by tracking work scheduled and what was actually accomplished. This information can contribute to PMOs in future planning providing better predictors for software availability and level of quality.

4.1 THE CUBIC CM PROCESS

Figure 4-1 illustrates the overall CM process and identifies how the tools and procedures used by CM work together to build quality software. This process results in high quality software for the over 15,000 users of CM supported programs at 200+ locations throughout the world.

The CM process begins and ends with the user. There are nine steps that are taken with every software release change.

1. Requirements Specification
2. Requirements Transmission
3. Status/Accounting
4. Verification
5. Development
6. Validation Testing
7. Certification Testing
8. Software Update
9. Requirements Evaluation

These are employed with varying degree of rigor, dependent on the scope of the change. The CM process is flexible allowing informed PMOs to make decisions on when expediting the process

will not add undo risk to the overall success of the program. The process also supports PMOs in identifying when full execution of the process is critical to mitigate risk to the Operational User of the Software Release.

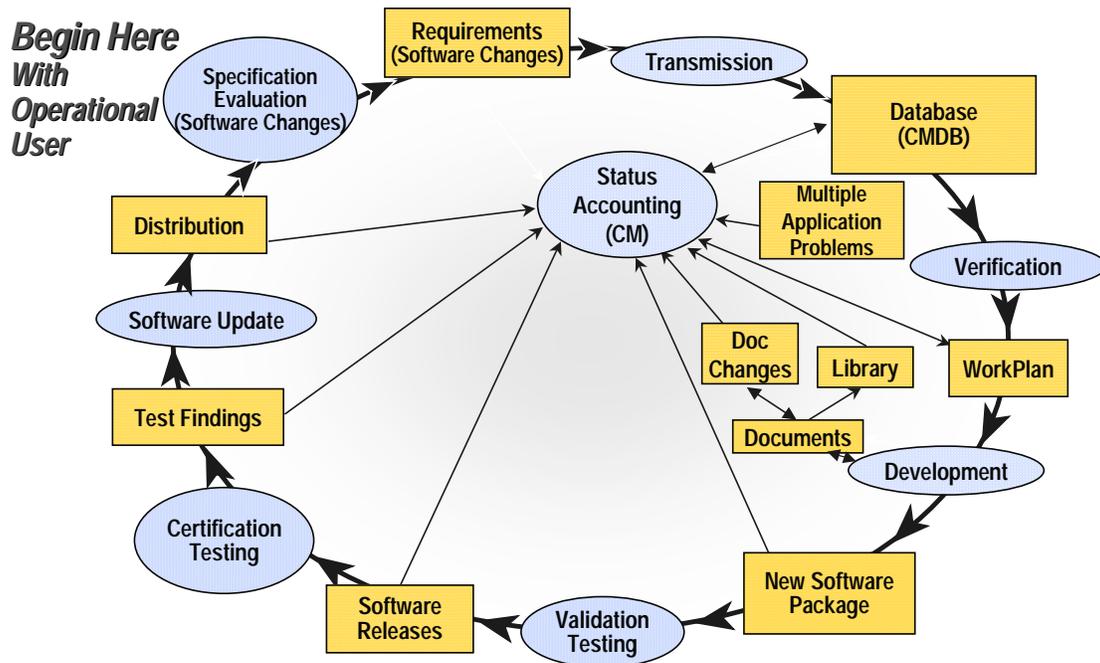


Figure 4-1 CUBIC CM Process

Software users provide requirements specifications (i.e. what the software must do to support their mission) at the beginning of the process and end the process by evaluating how well the software does it job (i.e. identifying new or missing requirements). (Steps 1 and 9)

These specifications and evaluations are provided in the form of PR/CRs, which are actually requirements for changes to the software package. These are transmitted (Step 2) into the CMDB. Details on submitting PR/CRs and other configuration items are available in Sections Seven through Section Eleven of this document.

Status Accounting (Step 3) processes all CM configuration items (DRRs, PR/CRs, MAPs, Workplans, etc.). Status Accounting oversees all aspects of PR/CR and other change control items' processing to ensure that no requirements are lost, misinterpreted or developed without proper CM and Quality Assurance (QA) documentation. The information tracked through the Status Accounting process facilitates the Configuration Control Management structure described in Section Five of this CM Plan.

During the Verification Process (Step 4) the PMO evaluates the requirement (PR/CR) to determine that a software change is necessary and identify if other programs are affected by the proposed change. Workplans provide critical information on the scope of the proposed change and the level of effort required in implementing the change.

Development activities (Step 5) include updates to software and documentation. Development activities begin only upon PM approval of a Workplan provided by the software developer. The CUBIC CM process does not replace the CM activities required to manage the development contractor's software engineering activities. Developers will use various tools to check software in/out for coders to work on and update associated documentation.

The resultant changes are packaged into a Software Release that will be Validated (Step 6) by the PMO and developers to ensure that requirements are met. The Software Release is then submitted to Certification Testing where independent testers such as the JITF, JITC and Security assess the Software Release.

Formal Certification Testing (Step 7) is required for most CUBIC programs prior to fielding a software release, since a majority of CM supported programs are used by the Department of Defense Intelligence Information System (DODIIS). Additional information on the Certification Process can be found in Section Six of this document and in the *DODIIS Instructions* and *DODIIS Test and Evaluation Policy* Document as noted in Section Two. Test Findings are generated and, if required, the Software Release is updated prior to Distribution (Step 8) of the new Software Release to the field.

The process ends with Users evaluating the Software Release to determine if requirements are met and identifying new software changes/requirements (PR/CRs). (Step 9)

5. CONFIGURATION CONTROL MANAGEMENT STRUCTURE

Change control management is designed to facilitate decision making and information flow between affected individuals and organizations. These processes and services promote issue resolution at the lowest effective level. Clear lines of responsibility for change encourage continual improvement and increase the productivity of supported programs.

5.1 PROGRAM MANAGEMENT OFFICE

The PMO identifies and evaluates advanced concepts and initiatives for the software and provides research, development, and enhancement engineering for Preplanned Product Improvement (P³I) for software in accordance with published standards.

In addition, the PMO is also responsible for all aspects of CM for the program. All PRs/CRs and Requirements, DRRs, documentation, AIs, user profiles, and software release information is logged and monitored using automated tools. The PMO also provides QA, ensuring that the software and documentation conform to established requirements, overseeing all aspects of testing, development and software operations.

5.2 PROGRAM CONFIGURATION CONTROL BOARD (CCB)

The purpose of a program CCB, with an executive manager as chairperson, is to provide a focus for change control. Meetings are called as required. Figure 5-1 illustrates the relationship between the PMO, CCB, developer, users, and executive managers and the flow of information, and resolution paths between these groups.

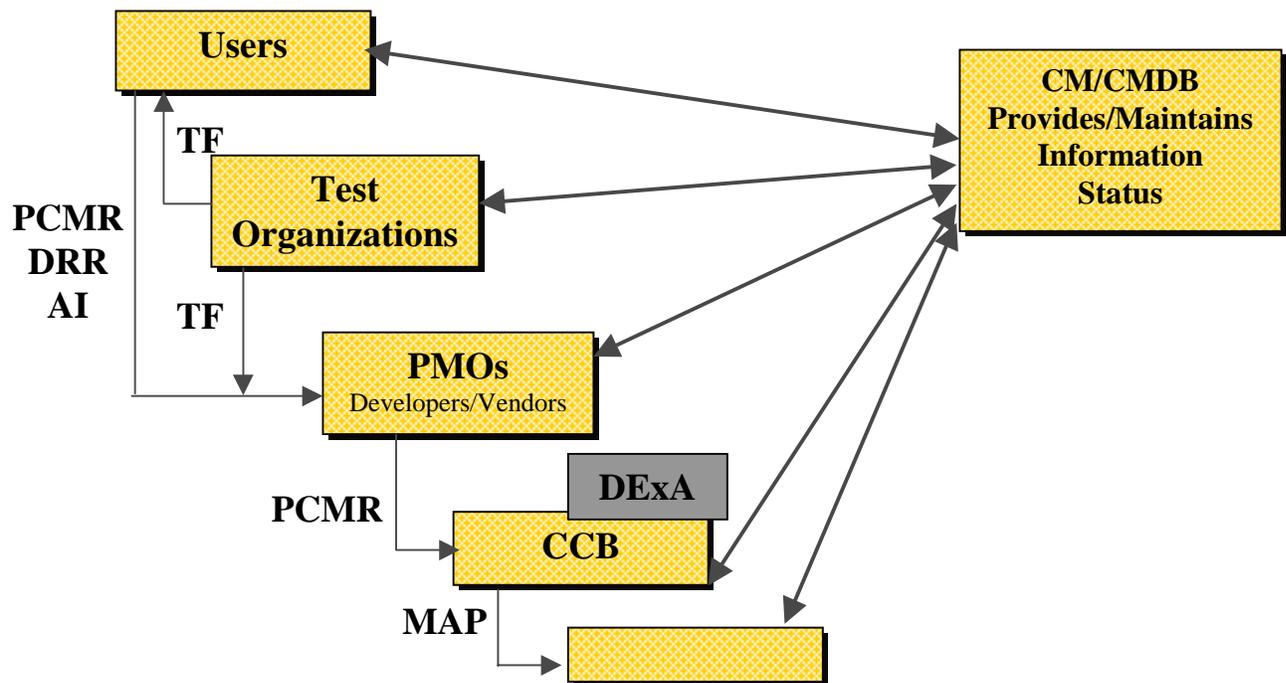


Figure 5-1 Configuration Control Management Structure and Information Flow

5.2.1 Program CCB Responsibilities

The Program CCB is usually responsible for the following activities:

- Enforces policy for life cycle management of the program.
- Determines the program's response to MAPs submitted by other CUBIC programs.
- Approves/disapproves and prioritizes all PRs/CRs by user survey with the final decision made by the CCB Chairperson. PRs are voted on at the discretion of the individual programs.
- Evaluates if approved CRs will initiate a MAP due to impact on interfaces or infrastructure.
- Approves/disapproves completed program MAP packages. Program approved MAP packages that have cost and schedule impact to the intelligence community are forwarded to the CUBIC CCB (CCCB) for approval/disapproval. Approved MAPs with no cost and schedule impact are implemented at the program manager's discretion, in coordination with the executive manager.

5.2.2 Program CCB Attendees

- Program executive manager
- Program manager (PM) or PMO representative
- AFRL or ESC Division Chief
- Site representatives
- Developer(s)
- Program support contractors, i.e., System Engineering and Technical Assistance (SETA), Independent Verification and Validation (IV&V), Technical and Engineering Management Support (TEMS)
- CM representative (AFRL)
- CUBIC executive manager (AC2ISRC/A-2)

5.2.3 Program CCB Agenda

CCB agendas are mandatory. The following is a suggested format:

- Old action item status review.
- Review of program interface matrices.
- Present and discuss new issues including status reviews of MAPs and CRs.
- Present and discuss recommendations.
- Evaluate and determine responses to MAPs submitted by other programs.

- Approval/disapproval of changes to baseline, e.g., CRs and program initiated MAPs.
- Review of decisions and action items.

5.2.4 Program CCB Minutes

Minutes of the proceedings, including a list of action items, will be taken by a representative of the program and submitted to the chairperson for approval within 10 working days after the meeting. Distribution to all attendees will occur within five (5) working days of the chairperson's approval. Program Manager's and Chairperson's signatures are required on minutes.

5.3 CUBIC CONFIGURATION CONTROL BOARD

The purpose of the CCCB, with the AC2ISRC/A-2, formerly 497IOG/IND, as chairperson, is to address problems outside the purview of the Program CCBs, problems unsolvable by the individual Program CCBs, and those problems and changes which affect multiple applications. A charter will be developed for the CCCB. The CCCB provides oversight for programs supported by CM and ensures that common problems are solved by common solutions in keeping with open systems philosophy. This is the highest level of oversight in the CM process.

5.3.1 CCCB Responsibilities

- Makes decisions on issues beyond the scope of Program CCBs, yet within the scope of the overall CUBIC program.
- Approves or disapproves all interfaces, and related issues, to or between supported programs, except those interfaces determined by the CCCB to be command specific.
- Approves or disapproves MAPs beyond the scope of the affected supported program CCBs.
- Forwards MAPs beyond the scope of the CCCB to the appropriate organizations.
- Reviews all program CCB minutes and reports.

5.3.2 CCCB Attendees

- Air Force Research Laboratory Information Handling Branch Chief
- CM representative (AFRL)
- CUBIC executive manager (AC2ISRC/A-2)
- Program managers or PMO representative
- Program executive manager
- Security representative(s)
- Training representative(s)

5.3.3 CCCB Agenda

CCCB agendas are mandatory. The following is a suggested format:

- Old action item status review.
- Present and discuss new issues including status reviews of MAPs.
- Present and discuss recommendations.
- Approval/disapproval of MAPs.
- Review of decisions and action items.

5.3.4 CCCB Minutes

Minutes of the proceedings, including a list of action items, will be taken by a representative of the CUBIC executive manager, such as a support contractor, and submitted to the chairperson for approval within 10 working days after the meeting. Distribution to all attendees will occur within five (5) working days of the chairperson's approval. The Chairperson's and Information Handling Branch Chief's signatures are required on the minutes.

6. CONFIGURATION MANAGEMENT ACTIVITIES

This section describes CM activities and its function of tracking necessary life cycle information for a program. These activities are the key to the organized effective implementation of CM in the software development process:

- Product Baseline Version Identification
- Document/Media Identification
- Configuration Control
- Configuration Status Accounting
- Configuration Installation & Testing
- Metrics
- Quality Assurance
- Reviews and Audits

6.1 PRODUCT BASELINE VERSION IDENTIFICATION

Software baselines and releases will be identified with a designator comprised of an integer to provide sequential numbering and decimal numbers indicating revision or version level. The first operational baseline version of the software will be identified with the designator 1.0.

A major release, e.g., 1.0 to 2.0, indicates a significant change in the architecture or operation of the application. A “rough rule of thumb” for PMOs to use for a significant change is 30 percent of the baseline changes. A minor release of a software version is indicated by a change in the decimal number, e.g., 1.0 to 1.1. A minor release includes new features but the fundamental architecture remains unchanged. A maintenance release or patch, e.g., 1.0 to 1.0.1, indicates new features may have been added, but the emphasis is on optimization, feature enhancements, or modifications to improve stability and usability. All release numbers are determined by the PMO.

The SVD will identify the exact version and contents of the software release package. As a minimum, the SVD will contain the following information:

- Release document identification
- Inventory and description of release package contents
- PR/CR change summary and notes
- Interface compatibility

6.2 DOCUMENT/MEDIA IDENTIFICATION

All documents/media stored in the program's technical library will be assigned control numbers as illustrated in Figure 6-1. It is recommended that all program managers include this numbering schema as part of their contract requirements.

IESS-3.0-DBDD-03MAR97

Figure 6-1 Document Media Number

The control number consists of four mandatory fields. The composition of the control number is as follows:

- The first set of letters comprise the application acronym, in this case the application is Imagery Exploitation Support System (IESS).
- The second set of digits indicates the software version number with which the document/media is associated, in this case the version number is 3.0.
- The third set of characters identifies the item in an acronym format. For example, "DBDD" indicates that the item is a Database Design Document (DBDD). An optional set of characters identifies the Configuration Software Control Item (CSCI) indicator and is used as required. If a virgule ("/") follows the CSCI indicator with a numeral, a volume is indicated. The volume indicator will always follow a virgule whether located after the document type or after the CSCI indicator, if one exists.
- The fourth set of digits and letters indicate the date of the document/media in the format DDMMYY, as seen in the example "03MAR97", i.e., March 3, 1997.

The PMO will apply appropriate Government or standard commercial labels on all media. These labels should show the application name, media title, date of media, classification, contractor name (if appropriate), and contractor media tracking number (if appropriate).

6.3 CONFIGURATION CONTROL

Configuration control encompasses the process of authenticating configuration identification documents and the processes by which changes are systematically requested, evaluated, classified, approved/disapproved, and subsequently implemented. Once the application components have been identified, configuration control measures assist in ensuring the established identification and allocation are preserved and no unauthorized changes are made to the application. Configuration control requires that changes to controlled configuration items (CIs) be submitted, evaluated, approved or disapproved, implemented, verified, and released according to established CM procedures. This aids in keeping configuration identification current and provides a reliable reference point for subsequent changes and development efforts. Once a baseline is established, the CIs constituting that baseline are subject to change control by the PMO.

6.3.1 Flow of Configuration Control

Configuration control calls for the documentation of all proposed changes; evaluation of these changes for cost, schedule, document and design impact, and orderly implementation of changes. The configuration control process consists of three (3) basic steps:

- *Step One - Initiation of the Change.* Proposed changes to the baseline are recorded as PRs/CRs/DRRs and assigned a unique identifier.

- *Step Two - Investigation and Review of Proposed Change.* The PMO validates every PR/CR/DRR. If the proposed PR/CR/DRR affects interfaces to other programs or external systems, the PMO will initiate a MAP to determine the impact to other PMOs and coordinate the proposed change. (PR/CR/MAP verification is discussed in Section Eight and DRRs in Section Nine.)
- *Step Three - Implementation of Approved Changes.* Approved changes are scheduled for incorporation and implemented based upon priorities established by the executive manager and engineering considerations.

6.3.2 Configuration Control Documentation

All proposed changes to the software and documentation will be submitted for consideration using the approved change control reporting formats. The following forms will be used to record and track changes:

Deviation: A configuration deviation is a specific written authorization granted prior to the manufacture of a computer program to depart from a particular performance or design requirement of a specification, drawing, or other document for a specific number of units or period of time. A deviation does not require revision of the applicable specification or drawing. A deviation may be converted to an Engineering Change Proposal (ECP) if it is determined that the change should be permanent. Deviations should not be submitted or authorized that affect service operation, logistics, interoperability, or maintenance. PMOs, in coordination with their program executive manager, approve or disapprove deviations. If the nature of these changes affect other programs or interfaces, the PMO should determine if a MAP is required.

DRR: A DRR is used during document reviews, design reviews, and testing to record problems found in any program documentation. Any participant in the review of the documentation may initiate DRRs. They are tracked by the PMO via CM to ensure required changes are incorporated into the documentation. DRRs are discussed fully in Section Nine.

ECP: A contract ECP can be used, depending upon the contract, to request an alteration in the configuration of an item that is delivered, to be delivered, or under development, after formal establishment of its configuration identification. An ECP shall be submitted to the Government by the developer on DD Form 1692, Engineering Change Proposal. Technical documentation and other information required for justification and explanation of the change may be provided in support of the ECP. ECPs are approved or disapproved by the PMO, in coordination with the executive manager. If the nature of these changes affect other programs or interfaces, the PMO should determine if a MAP is required.

MAP: A MAP coordinates changes that affect multiple applications. MAPs are logical extensions to the change control process and acknowledge that applications are no longer built as “stovepipes” and that all problems and changes cannot be resolved within a single PMO. The MAP process coordinates changes between CM supported programs and other DODIIS applications, non-DODIIS, and COTS products. MAPs are discussed in Section Eight.

PR/CR: A PR/CR is the means by which users report problems encountered with the baselined versions of the software, make suggestions as to how the software can be made more efficient, or

to request changes for enhancements. Section Eight contains detailed information on the processing of PRs/CRs.

SPR: A SPR will be used during design reviews and development contractor-run testing to record software errors. Any program participant may initiate SPRs. SPRs generated during formal government acceptance testing, e.g., in-plant, are monitored by the PMO to ensure necessary corrective action is accomplished and that changes are properly incorporated into the software. SPRs not resolved prior to acceptance of software will be converted into PRs/CRs for tracking.

TF: A TF is used by testers to report a software or document deficiency identified during formal testing, i.e., JITF, JITC, or Security, of software applications. Test Findings are discussed in Section Seven.

Waiver: A configuration waiver authorizes the acceptance of an item that varies from the baseline standards after final testing or at the start of production. PMOs, in coordination with their executive manager, approve or disapprove waivers. If the nature of these changes affect other programs or interfaces, the PMO should determine if a MAP is required.

6.3.3 Documentation Storage and Control

A hardcopy and softcopy of program documentation should be forwarded by the PMO for inclusion into the CM library, as required by the program contract. Upon receipt, documents are logged into the CMDB. The current and two previous versions are retained. This will be done for change pages as well. AF Form 310, Document Receipt and Destruction Certificate, controls classified documentation. The receipt, storage, reproduction, and destruction of classified documents will be in accordance with established government security procedures.

All documentation (new or revised) will be reviewed by the PMO in accordance with the documentation review procedures outlined in Section Nine. Disapproved documentation will be returned to the development contractor, along with DRRs, for incorporation into the next revision.

6.3.4 Document Reproduction Requests

Requests for copies of library documentation will be based upon need-to-know as determined by the PMO in coordination with their executive manager. An approved documentation set will be sent to users with the release of a new baseline.

6.3.5 Software Storage and Control

The development contractor stores all software under development for a given program, as part of the development configuration. CM will store PMO accepted software as the product baseline. As a minimum, a master copy of all baselines currently in operational use should be stored in CM.

6.3.6 Software Reproduction Requests

CM offers CD, 4mm and 8mm tape, and diskette duplication. CM, upon request, will duplicate and distribute program-related items based on approval by the PMO in coordination with their executive manager. Copies of COTS products can be provided with appropriate proof of

licensing. Backups of program related software can also be accomplished with minimal advance notice.

6.4 CONFIGURATION STATUS ACCOUNTING

Status accounting oversees all aspects of CM processing and requires all participants to follow a pre-established workflow. Adherence to this workflow will ensure that status accounting products (PR/CR/Requirement/DRR/AI/Library database, acknowledgment messages, and status reporting) will be accurate and timely. The majority of information regarding the life cycle of the configurable items are logged and tracked using the CMDB for each program. Concurrence information and cross-references to related items are also logged.

6.4.1 CMDB

The CMDB is the automated status accounting tool used to record and report information. The CMDB is a relational database with a web browser front-end. There are two servers, one on the Internet and the other on Intelink. Access is controlled by user accounts and passwords. Users can view and submit information directly from/to the database. Hardcopy reports can be provided, upon request to sites not having Internet/Intelink access.

The CMDB provides current real-time information to the PMO regarding the status of the program. The CMDB also provides insight into the application in order to control change and assist in the production of the highest quality software products possible.

The CMDB provides process support for the following items:

- PR/CRs
- Workplans
- AIs
- DRRs
- MAPs
- Test Findings
- Requirements
- Documentation
- Software
- Software Release Information
- CMDB User Profiles
- Place (or Site)
- POCs

For additional information on these components, refer to Sections Seven through Eleven. Further information on the CMDB can be obtained in the CMDB Users Manual available upon request from the CM library. The CMDB System Software Specification provides a comprehensive listing of current requirements, both implemented and planned. Additional planned functionality for the CMDB includes adding Distribution and CDRL tracking capabilities to a future version of the software. The CMDB is managed using the CUBIC process

and any suggestions for improvements or error correction can be documented using the processes described in the CM Plan.

6.4.2 CMDB Access

New users are required to complete a profile detailing their address, phone, facsimile, unclassified e-mail, POC and AIG status, and programs of interest. Notification of new profiles will be sent to the identified PMO(s) for approval for access to program information and to CM for account creation. Upon approval and account creation, each user will be assigned a login id and password. The user will be e-mailed or mailed their account Login IDs and assigned an authorization number that CM will track. The user then calls CM with this authorization number for verification and CM will provide a unique password.

Profile information is an essential part of the CMDB. Accounts are created according to roles and permissions. A user's login and personal information will be maintained through his/her individual profile.

Each profiled user of the CMDB will be assigned a specific role for each application the individual has subscribed to within their profile. Specialized access (Read, Add, Modify, or Delete) to the database will be determined by the permissions assigned to that specific role. Only the PMO can assign these roles based on what the user's need to know.

6.5 METRICS

CM will use information stored in the CMDB to generate metrics to provide insight into CM processes, assess variation between releases, and generate information on the software development process. Examples of metrics include, but are not limited to:

- Length of time a PR/CR remains open in the system.
- Length of time from workplan approval to closure for PRs/CRs.
- Installation rate of new releases.

This information will be provided, upon request, to the PMO, executive manager, CCCB Chair, etc.

6.6 QUALITY ASSURANCE

Many times software development problems can be attributed to the lack of software QA controls. QA, as applied to software, is the systematic evaluation of software and related documentation by an independent person within the program's software development team. The implementation of a tailored QA effort to support software development and maintenance should result in a more orderly process, and permit errors to be found early in the development cycle.

Software development that does not demonstrate the required maturity or completeness as judged by QA controls will be stopped until the documented discrepancies (PRs/CRs, SPRs, TFs, and DRRs) are corrected and incorporated into the development. SPR, TF, and PR/CR QA is a planned set of maintenance procedures, documentation, and actions necessary to provide adequate confidence that the software will perform satisfactorily in actual operations. For example, a software release with at least one Impact Code 1 PR or SPR discovered during in-

plant testing will not proceed to the next phase, e.g., JITF, JITC, or security accreditation. Proceeding would waste time and money since it is already known that the application cannot be used operationally. A major design flaw discovered during review means that the system should not enter the coding phase.

There is no software development strategy that suggests that QA must be performed only on programs during their original development phase. All programs generate SPRs, TFs, and PRs/CRs in which coding errors are discovered during development and operation, regardless of the age or complexity of the software version.

Implementing QA can result in a higher quality product, with greater visibility into the development process and a better understanding of user/customer needs. Although QA has been described as a formal process, it can be done informally by anyone with an objective, independent look at the software development for the program.

6.7 REVIEWS AND AUDITS

Program software, hardware, and documentation development will be monitored through reviews and audits in accordance with current standards listed in Section Two. During reviews and audits, application documentation will be evaluated to measure development or installation progress, ensure that program requirements are satisfied, and identify existing and potential problem areas. Software review and configuration audits verify that computer code meets functional and performance specified in the documentation. Hardware configuration audits verify functional and performance specifications and confirm that installations have followed accepted engineering practices.

In addition to formal audits of physical and functional baselines, all contributors to the program, e.g., Government representatives, development and IV&V contractors, and operational sites, should meet semi-annually to review the status and disposition of all open CM records. This activity ensures that status accounting information remains accurate and current.

6.7.1 DODIIS Certification Process

The DODIIS Certification Process, shown in Figure 6-2, performs an audit function for those programs requiring DODIIS certification. The Certification process ensures that developed software and documentation meets requirements. The DODIIS System Integration Management Office (SIMO) functions as the auditing organization evaluating input from various formal test organizations and the PMO. Hardware audits and inventories are completed by Service level SIMOs. The majority of programs supported by CM follow the procedures in the DODIIS Certification Process.

6.8 CONFIGURATION INSTALLATION AND TESTING

This DODIIS Certification process is based on DOD 5000.2-R Lifecycle Management and requires independent testing of software releases to ensure integration, interoperability and security. CM supported programs use SPRs, TFs, PRs/CRs, DRRs, and AIs to track problems with software and documentation identified in the course of independent testing. This process provides assurance that intelligence applications work as planned when they are delivered to

sites. Additional details on the DODIIS Certification Process can be found in the DODIIS Instructions and the Test and Evaluation Policy Guidance documentation (See Section Two).

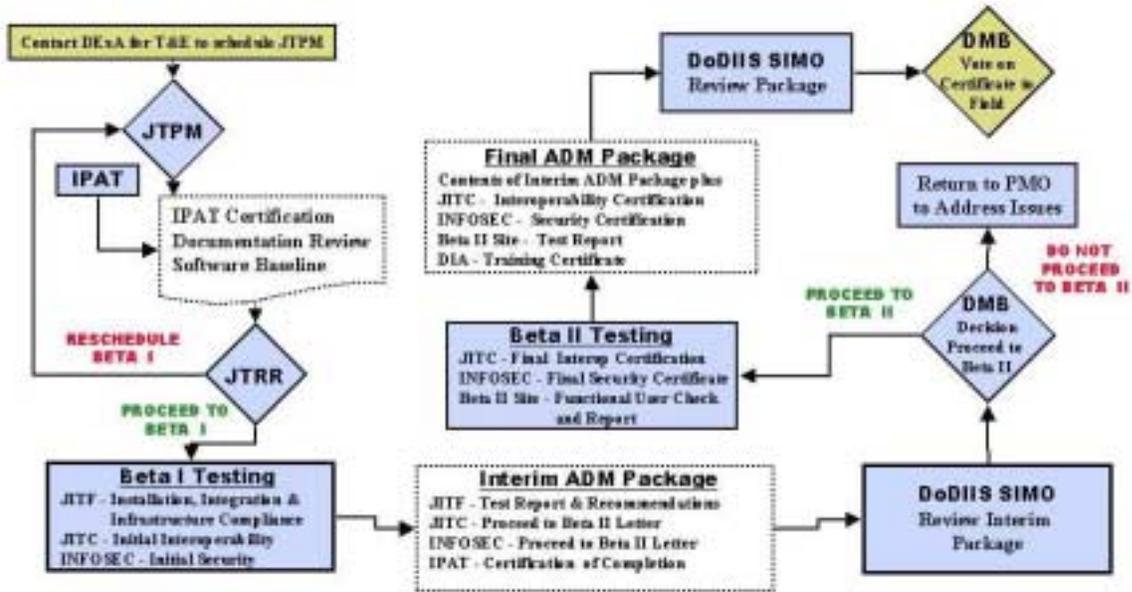


Figure 6-2 DODIIS Certification Process

CM or the developer, per contract requirements, will deliver approved release packages to the sites. The sites will install and test the release packages to validate successful installation and correction of the applicable PRs/CRs. The site is responsible for testing and validating the incorporation of PRs/CRs listed in the SVD. Refer to Section Eleven for more information on software releases.

7. TEST FINDING (TF) PROCESS

The objective of this process is to ensure that all findings discovered during formal testing, i.e., JITF, JITC, or Security are tracked and made available to the PMO, decision-makers and application users.

The TF process, as illustrated in Figure 7-1, provides coordination between all pertinent parties, ensuring that findings from formal testing are properly tracked and implemented. The following is the process in which TFs are identified, tracked, and disseminated.

- ❑ *Step 1* - A finding is identified during testing and submitted by the tester using the input forms in the CMDB. If the TF is classified, the tester must submit it by using the input forms on Intelink.
- ❑ *Step 2* – Once testing is complete, the Test Director and CM review the TFs for accuracy, completeness, and duplication.
- ❑ *Step 3*– If the TF was written against an application that is supported by CUBIC, the document and software findings discovered in test can be cross referenced to a PR/CR or DRR at the application PM’s direction.
- ❑ *Step 4* – Prior to the next test of the application, all open TFs are reviewed and statuses are updated, as necessary.
- ❑ *Step 5* – During the next test, the previously identified TFs are validated to ensure implementation. If the TFs have been implemented, they are closed by the testing organization. Any new findings are submitted. At this point, the process starts again.

7.1 IMPACT CODE DESCRIPTIONS

Each TF identified by the various test organizations are assigned impact codes that identify the anticipated impact to operational users or the impact on the test process, as applicable. In general, the impact code definitions are similar, but specific definitions exist for JITF and Security testing.

7.1.1 JITF

The following defines the four types of impact codes used by the JITF:

- *Impact Code 1* - A finding that, without resolution, either
 - a. Prevents either the application under evaluation or another application or component of the infrastructure from operating properly;
 - b. Creates a security vulnerability in the application or site architecture that can be exploited by a general user without taking advantage of other vulnerabilities or capabilities; or
 - c. Seriously increases the level of effort of site personnel to manage and/or use the application under evaluation or other applications.

An Impact Code 1 finding is assigned if the application baseline must be changed in order to continue testing, if the resolution requires an excessive level of effort, if Configuration and Installation Documentation does not support the installation and integration process. An Impact Code 1 can also be assigned if the resolution introduces additional problems in the installation or operation of the application.

The level of effort is a key determinant for Impact Code 1 findings. The time or expertise that is required to install, manage, or use the application cannot exceed what is reasonably expected for an application.

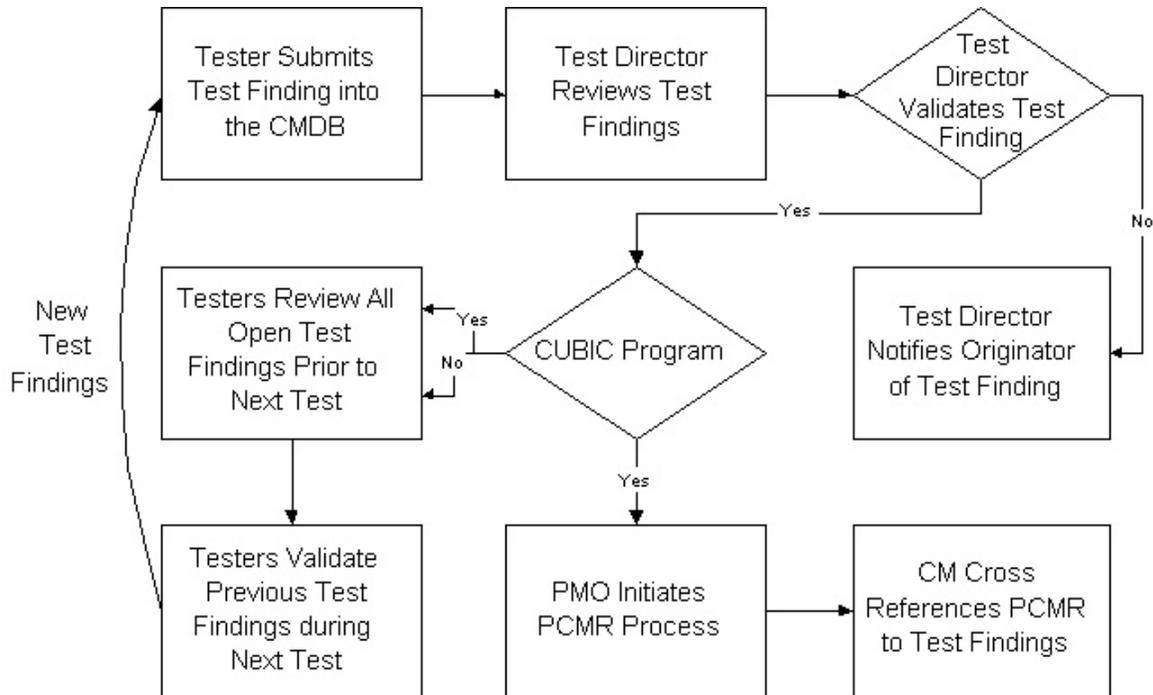


Figure 7-1 Test Finding Process

- *Impact Code 2* – A finding that, without resolution,
 - a. has a significant effect on the operation of either the mission application or on another application or component of the infrastructure; or
 - b. creates a security vulnerability in the application or site architecture that could be exploited by a general user only if the user is able to take advantage of other vulnerabilities or capabilities not typically available to him or her.

The finding can be temporarily resolved by a change in procedure or configuration. The successful resolution requires technical expertise that is not expected of general users, or the resolution requires a significant level of effort by site administrators. The resolution does not cause significant delay in integration testing; instead, it can be proposed and evaluated during integration testing at the JITF.

An Impact Code 2 problem may be elevated to an Impact Code 1 if proposed resolutions either do not work successfully or produce additional Impact Code 2 and 3 findings.

- *Impact Code 3* - A finding that, without resolution, has a significant effect on the operation of either the application under evaluation or on another application or component of the infrastructure. The finding can be temporarily resolved by a change in procedure or configuration. The successful resolution does not require technical expertise that is not expected of general users, or the resolution does not require a significant level of effort by site administrators. The resolution does not cause significant delay in integration testing; instead, it can be proposed and evaluated during integration testing at the JITF.

Impact Code 3 findings do not cause integration test failure, but the accumulation of Impact Code 3 findings may affect the JITF's "go/no go" recommendation.

- *Impact Code 4* - A finding that does not significantly affect the operation of the application under evaluation or another application or component of the infrastructure. The finding can be resolved by a workaround that can be implemented as a change in procedure or configuration during integration testing without a significant level of effort, or the finding can be left as is. Even though the finding has some affect on the configuration or operation of the mission application or of other components of the site architecture, the general user will be able to perform mission functions, and the administrator will be able to manage the mission application. Findings in this category are of lesser importance, but the accumulation of Impact Code 4 findings may affect the JITF's "go/no go" recommendation.

7.1.2 Security

TFs generated during security testing use four Category codes. Any security test finding that identifies a vulnerability to the system is considered classified.

The following defines the definition for security testing impact codes or Category Codes:

- Security Category Code I - A significant security finding which must be fixed prior to operational use.
- Security Category Code II - A security related finding which must be fixed within a specific time period in order for approval to be granted.
- Security Category Code III - A security relevant recommendation for which implementation is optional.
- Security Category Code IV - A non-security relevant recommendation for which implementation is optional.

8. PROBLEM REPORT/CHANGE REQUEST/MULTIPLE APPLICATION PROBLEM/REQUIREMENT (PCMR) PROCESS

The objective of the PR/CR/MAP/Requirement (PCMR) process is to provide a methodology based upon sound engineering principles that will close the user-developer communications gap, place software development control firmly in the hands of the procuring agency, and produce computer software products that are acceptable in the eyes of the user. The PCMR engineering management process fully encompasses all software development functions from requirement definition through test and integration.

The PCMR procedure ensures the coordinated participation of all interested members of the organization throughout the software development process. User requirements are identified and, through the detailed procedures, tracked and monitored by CM using the CMDB. This approach reduces technical, schedule, and cost risks.

8.1 CM RESPONSIBILITIES

CM is the central repository for all PCMRs submitted against supported applications' software. Once the software baseline has completed the approval process, all software problems and proposed changes will be submitted, evaluated, approved or disapproved, implemented, verified, and released according to these PCMR procedures. This scheme will ensure that the configuration identification for the software is current and that no unauthorized changes are incorporated.

PRs will be used to record discrepancies found in the operational software. CRs will be used to request changes and/or enhancements to the operational baseline. MAPs will be used for PRs/CRs that affect multiple CM-supported applications or CM supported applications and other DODIIS applications, non-DODIIS, and COTS products. CM provides the program CCB and CCCB with information necessary for the decision making process. This process ensures that changes to the application are controlled and PMOs and users know the costs and benefits.

The MAP process enhances interoperability by maintaining interfaces, as defined in Interface Control Documents (ICDs), between supported programs, and coordinating advances in infrastructure support products. The process also provides a unified voice for the intelligence community to address problems and required changes to COTS products, ensuring that commercial vendors appreciate the impact to the complete customer base.

ICDs are required documents under the DODIIS Certification Process. An ICD Data Item Description (DID) was initially approved for CUBIC and can be used as a guideline for PMOs requiring ICDs. A copy of the DID can be obtained from CM.

Requirements are information maintained by CM to track program level requirements for future versions. These requirements translate into program specifications. PR/CRs can be cross-referenced to Requirements when these are maintained in the CMDB.

8.2 PCMR PROCESS DESCRIPTION

The PCMR process, as illustrated in Figure 8-1, starts and ends with the user and involves the entire user community. The procedure for processing PCMRs provides coordination between all

of the players, ensuring that user requirements are not lost and are properly implemented. The steps to this process are described in the following paragraphs.

□ *Step 1* – The PMO, executive manager, site, or the development contractor can submit a PCMR by using the PCMR (New Record) form in the CMDB at any time during the life cycle of the program. Users who do not have Internet/Intelink access can use E-mail, message traffic via the AUTODIN communications network, facsimile, or regular mail. If the PCMR is classified, the user must submit it by using the CMDB PCMR (New Record) form on Intelink or other appropriate means for transmitting classified information. If the PCMR is an Impact Code 1, follow the procedures outlined in Section 8.2.3. Impact Code descriptions are detailed in Section 8.2.1.

□ *Step 2* - All PCMRs are sent to the PMO for evaluation. CM will verify that all necessary information has been included in the form or message and that the problem exists in a currently approved baseline.

- If additional information is needed, the originator is contacted to request further details or clarification of the problem or new requirement. Notification is sent to the originator to verify that the PCMR has been received. To eliminate duplication and time-consuming evaluation, sites should review PCMRs before submitting a new PCMR. A site that is experiencing the same problem submitted by another site should add a concurrence message via the Notebook option in the CMDB to the PCMR, referencing any additional information regarding the problem they are experiencing at their site. Concurrence messages play a significant role for successful software maintenance efforts on any PCMR. This provides the PMO and other users with essential information on the scope and criticality of the problem or requested change.

□ *Step 3* - The PMO will verify the PCMR by reproducing the problem on the program's current baseline or through analysis. This verification is performed/directed by the PMO engineer and could be performed anywhere the baseline exists other than the originating site. Information identified during the verification activity can be forwarded to the developer to assist in the creation of a workplan. If the PCMR is disapproved, the originator and any other pertinent agencies will be notified via the CMDB. Valid PCMRs will be examined to determine that the impact code is correct and to verify that the PCMR does not duplicate work already in progress. Duplicate PCMRs will reference the PR already being worked. If additional information is provided in the duplicate PCMR that will assist in the correction of the original PR, it will be noted. The same action will take place with any concurrence messages from other sites. CRs will be evaluated and approved or disapproved by the PMO upon coordination with the executive manager and program CCB procedures. The PMO will determine if the PR/CR affects another application; if it does a MAP will be initiated.

□ *Step 4* - The PMO will determine if a workplan should be requested from the development contractor. A workplan outlines the estimated time, cost, schedule, and design approach to implement an approved PCMR. It also includes a history of all affected software

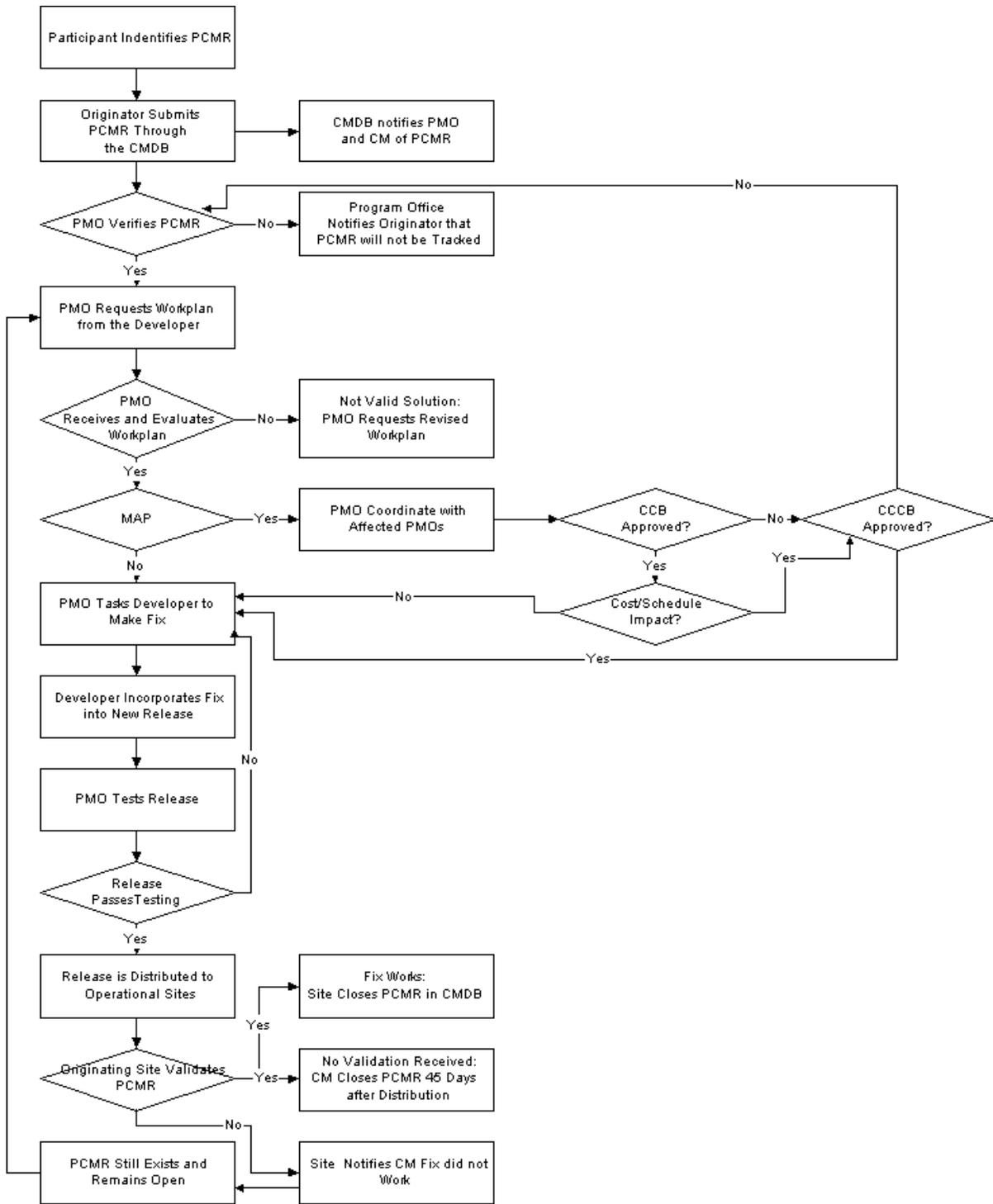


Figure 8-1 PR/CR Process

modules and documents. This ensures that the costs and benefits of problem fixes and enhancements to the application are known by program management and are considered when determining changes to software baselines.

Requesting a workplan from the development contractor is accomplished via a notification letter generated by CM upon direction of the PMO or through notification via the CMDB.

- ❑ *Step 5* - If upon developer's review the problem is valid it is determined that the change will other programs, a MAP record will be entered and all affected PMOs will be notified. PMOs will review the MAP and provide an impact analysis or workplan.
- ❑ *Step 6* - The development contractor submits the workplan via the CMDB to the PMO for review within 14 days of receipt of notification. The amount of time the development contractor has to submit a workplan is often specified in the contract. The 14 days is to be used as a guideline. The PMO, coordinating with the executive manager, may decide upon review of the workplan to discontinue the effort based upon a variety of factors, e.g., funding or time constraints and situations overtaken by other events, etc. If this occurs, the PCMR will be closed and all pertinent agencies will be notified. If the workplan is not a valid solution, a revised workplan will be requested or the PCMR will be re-evaluated by the PMO and, depending upon the circumstances could be closed.
- ❑ *Step 7* - The development contractor is directed via a tasking letter sent by CM or notification via the CMDB upon PMO approval of the workplan, to implement the change as described.
- *For MAPS*: CM coordinates the MAP process with the PMOs, ensuring that all information is collected and presented at the next scheduled program CCB.
 - The program CCB reviews and approves or disapproves the MAP package. If approved and there are no cost and schedule impacts, the executive manager, in coordination with the PMO, determine the implementation strategy. If approved and there are cost and schedule impacts the MAP package is forwarded to the CCCB.
 - The CCCB approves or disapproves the MAP. If approved, the change is implemented as prescribed by the MAP package for each affected application. Disapproved MAPs are closed by CM or re-evaluated by repeating the MAP process. A MAP may require coordination by the CCCB with CCBs outside the programs supported by CUBIC prior to approval or disapproval of the requested change.
 - Based on the complexity of the MAP, the PMO may want to coordinate an Interface Control Working Group (ICWG) comprised of members from the affected programs to assist in working out the MAP issues.
- ❑ *Step 8* - The development contractor completes the change and incorporates it into the next release.
- ❑ *Step 9* - The release is tested.
- ❑ *Step 10* - The approved release is then distributed to the sites along with a SVD and other application documentation as required. The sites are then responsible for referencing the SVD and verifying any PR/CR listed as originating from their site and ensuring it has been incorporated. The sites need to update the status, fixed or problem still exists, in the CMDB within 45 days of the release with their responses.

- ❑ *Step 11* - If no validation information is provided by the site after 45 days, the PRs/CRs listed in the SVD will be closed by CM upon direction of the PMO and notification will be sent to the pertinent agencies. Sites can request to extend the 45-day deadline or the PMO can designate an alternate timeframe.
- ❑ *Step 12* - If the PR/CR was not implemented as stated in the SVD, the PR/CR remains open. At this point the cycle begins again at Step 4.

8.2.1 Impact Code Descriptions

The following paragraphs define the five types of impact codes:

- *Impact Code 1*
 - a. Prevents the accomplishment of an essential capability.
 - b. Jeopardizes safety, security, or other requirement designated "critical".

Impact Code 1 problem reports are considered classified. See Section 8.2.3 on how to enter an Impact 1 PR.

- *Impact Code 2*
 - a. Adversely affects the accomplishment of an essential capability and no work around solution is known.
 - b. Adversely affects technical, cost, or schedule risks to the project or to life cycle support of the system, and no work-around solution is known.
- *Impact Code 3*
 - a. Adversely affects the accomplishment of an essential capability but a work-around solution is known.
 - b. Adversely affects technical, cost, or schedule risks to the project or to life cycle support of the system, but a work-around solution is known.
- *Impact Code 4*
 - a. Results in user/operator inconvenience or annoyance but does not affect a required operational or mission-essential capability.
 - b. Result in inconvenience or annoyance for development or maintenance personnel but does not prevent the accomplishment of the responsibilities of those personnel.
- *Impact Code 5*
 - a. Any other effect.

8.2.2 Classification Issues

Classified PR/CRs must be submitted via secure methods. For security reasons, classified PRs/CRs will be maintained on the CMDB Intelink server. The only information that is

guaranteed to be unclassified, and will be maintained on the CMDB Internet server, is the PCMR number, date, type, impact code, base release number, originator name, phone, and status. A program engineer or manager will check the title and, if unclassified, it can be entered. The engineer or manager will select an unclassified title for entry into the CMDB if the original title is classified.

8.2.3 Submission of an Impact Code 1 PR

The following paragraphs describe the steps necessary in the life cycle of an Impact Code 1 PR. The process is illustrated in Figure 8-2.

- ❑ *Step 1* - The Impact Code 1 is identified from the originating site.
- ❑ *Step 2* - The site notifies the PMO via telephone stating that a Code 1 problem exists.
- ❑ *Step 3* - The site enters the PCMR into the CMDB on Intelink.
- ❑ *Step 4* - The PMO evaluates the problem to make a determination if the PR is an Impact Code 1 with coordination of the executive manager.

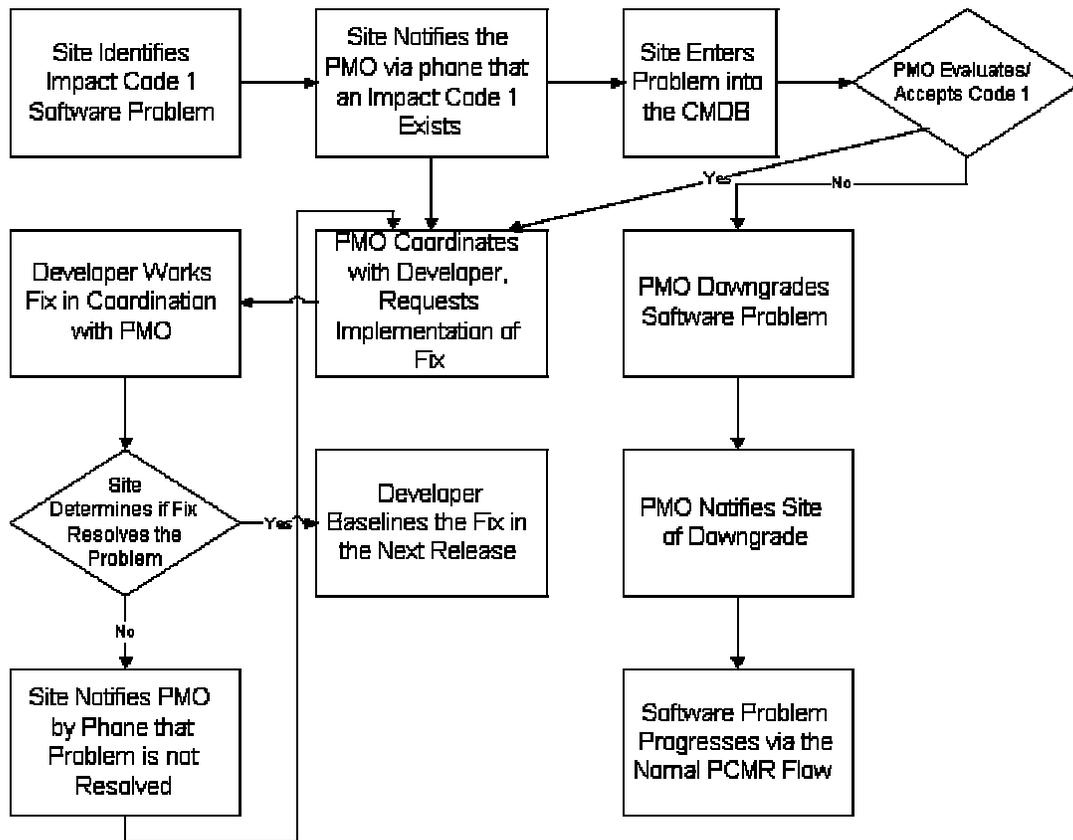


Figure 8-2 Impact Code 1 PR Process

- ❑ *Step 5* - If the PR is an Impact Code 1, the process will proceed to step 6. If it is not an Impact Code 1, it will be downgraded and the site will be notified by the PMO with the reason for the downgrade. The PR will progress from this point via normal PR workflow channels.
- ❑ *Step 6* - The PMO, or their representative, coordinates with the development contractor via phone or other immediate way of communication and requests work to begin on the fix, with precautions taken due to the classification of the issue.
- ❑ *Step 7* - The development contractor works the fix in coordination with the PMO and implements the fix at the affected site.
- ❑ *Step 8* - The originating site determines whether the fix resolves the problem and updates the CMDB with the results. If the fix resolved the problem, the process proceeds to step 9. If the fix did not resolve the problem, the PMO notifies the development contractor of the updated status and they rework the problem, as necessary.
- ❑ *Step 9* - The PMO will notify the development contractor and update the status.
- ❑ *Step 10* - The development contractor distributes the fix to other affected sites and baselines the fix in the next software release

8.2.4 PCMR Status Values

Each PCMR record has various states in its lifecycle, which are tracked in the CMDB. Figure 8-3 identifies the status values that can be assigned to a PCMR record from initial identification through closure. Not all PMOs use every available status.

Status Value	Role	Definition
Open	CM/PM, Originator	The PCMR has been identified and documented.
New	CM/PM	A flag status, which indicates that the PCMR has not been reviewed by the PM. This is the initial status of a PCMR for a PM.
In-Review	CM/PM	The PMO is reviewing the PCMR. The PCMR has not been validated as an actual software problem/requirement.
Accepted	CM/PM	The PCMR has been validated as a software problem/requirement. Open also has this status when used by the PM role.
Work Plan Requested (WP Requested)	CM/PM	The PM requests a Workplan from the Developer identifying the level of effort to implement the change requested in the PCMR.
WP Submitted	DEV	The Developer of the software submits a Workplan
WP in Review	CM/PM	The PM is reviewing the Workplan for acceptance
WP Rejected	CM/PM	The PM has rejected the proposal in the Workplan for implementing the PCMR.
Revised WP Requested	CM/PM	The PM requests a revised Workplan from the Developer.
Revised WP Submitted	DEV	The Developer submitted a revised workplan.
WP Accepted	CM/PM	The PM accepts the level of effort and solution proposed in the

		Workplan.
Revised WP Accepted	CM/PM	The PM has accepted the revised workplan.
Implemented	CM/PM	Identifies the PCMR as being incorporated into a software release, but not formally tested and closed.
Closed	CM/PM, DEV, Originator	<p>Closed means that work for the PCMR and all related activities are complete. The Closed status is used by each of the participants in the PCMR process, but has slightly different meanings.</p> <ul style="list-style-type: none"> • Developer – The correction for the PCMR has been incorporated into a software version that may or may not have been delivered to the Government. • PM – The PM has verified that the correction for the PCMR has been incorporated into a software version delivered to the Government. • User – The originator of the PCMR identifies that the PCMR has been corrected to their specifications.
Problem Still Exists	CM/PM, Originator	Identifies that the PCMR correction does not resolve the problem/requirement identified by the User after formal delivery of the software to operational sites.
Withdrawn	CM/PM, Originator	The PCMR is not a valid problem/requirements, is overcome by events, or is a duplicate of another PCMR.
Rejected	CM/PM	Identifies a PCMR that is considered valid by a User, but the PM does not identify the PCMR as requiring a software change.
Request Additional Information	CM/PM	Used by the PM to indicate that additional information is required from the originator or the Developer.
On-hold	CM/PM	Indicates the PCMR is in a hold status usually due to inadequate resource.

Figure 8-3 PR/CR Status Values

9. DOCUMENT MANAGEMENT PROCESS

CM provides complete library services and all documents and media received for supported programs are stored in the CM Library. These documents under go continual update and improvement. DRRs are a tool used within a program to evaluate, verify, and approve or disapprove the accuracy and adherence to standards for documentation releases. The document review process, as illustrated in Figure 9-1, ensures that all DRRs are implemented according to established CM procedures.

9.1 DRR PROCESS DESCRIPTION

- ❑ *Step 1* - The developer submits the documentation in draft form to CM via the PMO.
- ❑ *Step 2* - CM logs the documentation into the CMDB technical library, compiles any old DRRs that may have been written against the previous version of the document, and notifies appropriate technical personnel of the receipt of the revised documentation and the existence of any old DRRs.
- ❑ *Step 3* - Technical document review is coordinated by the PMO. Document reviews are generally initiated by QA personnel within the PMO, the development contractor, and/or any other requested reviewers. Specific users may be asked to comment on documents as well. Each review group has a designated DRR coordinator.
- ❑ *Step 4* - The reviewers assure that any old DRRs written against a previous version of the document have been incorporated, as well as writing up any new errors.
- ❑ *Step 5* - After all the comments are documented by the review group, they are submitted to the DRR coordinator who reviews them for completeness and duplicate comments. The coordinator then enters the DRRs into the CMDB using the DRR (New Record) form in the CMDB. If the DRR is classified, the DRR must be submitted through the DRR (New Document) form of the CMDB on Intelink. If a submitter does not have access to the database, hardcopies will be accepted.
- ❑ *Step 6* - CM initiates a transmittal letter upon direction of the PMO to notify the developer of DRRs against submitted documents. Developers are also notified of DRRs via the CMDB.
- ❑ *Step 7* - The development contractor reviews the comments and reworks the documentation as necessary before submitting a revised edition back to the PMO. The development contractor can use change bars when revising documentation to allow the reader to see what information has been updated.
- ❑ *Step 8* - Developers respond to the comments by entering a status and disposition in each DRR record.
- ❑ *Step 9* - PMO and any other original reviewers, if possible, review the revised documents along with the developer's disposition comments and decide whether the DRR has been satisfied. The PMO, depending upon the developer's comments, may close, withdraw, or hold the DRR open. Those held open will remain so until the developer forwards further revised

documentation and the DRR is deemed rectified by a Government reviewer. Disputes over Government validated DRRs must be resolved by the developer with the PMO prior to the release of the revised documentation.

- If the PMO does not approve the revised document, the open DRRs are returned to the developer and Step 7 begins again.

□ *Step 10* - Approved documentation is then designated for distribution as required to the operational sites.

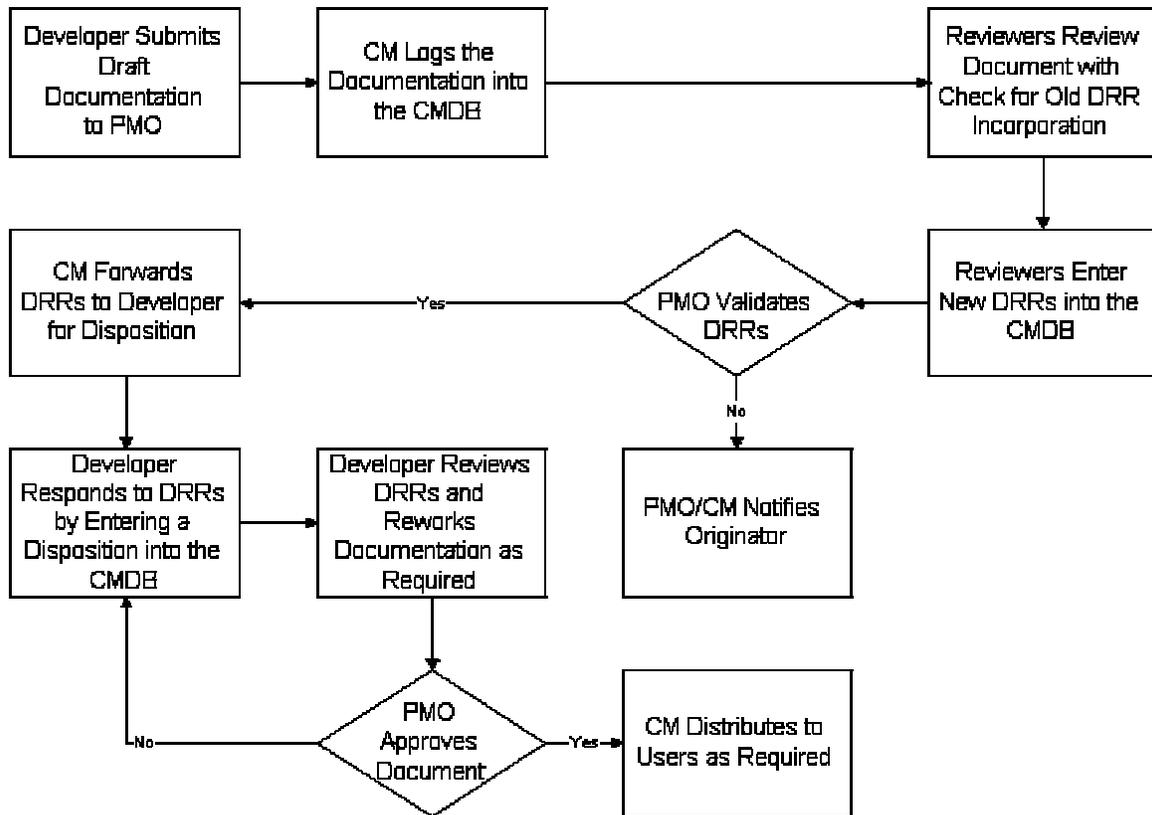


Figure 9-1 DRR Process

9.1.1 DRR Status Values

Each DRR record has various states in its lifecycle. Figure 9-2 identifies the status values that can be assigned to a DRR record from initial identification through closure.

Status Value	Role	Definition
Open	CM/PM, Originator	The DRR has been identified and documented. For the PM role this means the DRR has been accepted as a valid change required for the document.
New	CM/PM	A flag status, which indicates that the DRR has not been reviewed by the PM. This is the initial status of a DRR for a PM.
In-Review	CM/PM	The PMO is reviewing the DRR. The DRR has not been validated as an actual software problem/requirement.

In Dispute	CM/PM, DEV	Identifies a DRR which is being questioned by the PM and/or Developer as being a valid change to the document.
Closed	CM/PM, DEV, Originator	Closed means that work for the DRR and all related activities are complete. The Closed status is used by each of the participants in the DRR process, but has slightly different meanings. <ul style="list-style-type: none"> • Developer – The correction for the DRR has been incorporated into a document that may or may not have been delivered to the Government. • PM – The PM has verified that the correction for the DRR has been incorporated into a document delivered to the Government. • Originator – The originator of the DRR identifies that the PCMR has been corrected to their specifications.
Withdrawn	CM/PM, Originator	The DRR is not a valid change, is overcome by events, or is a duplicate of another DRR.
Rejected	CM/PM	Identifies a DRR that is considered valid by an Originator, but the PM does not identify the DRR as requiring a document change.
Request Additional Information	CM/PM, DEV	Used by the PM or Developer to indicate that additional information is required from the originator, the PMO or the Developer.
Revised	CM/PM, DEV	Identifies that there is a revision to a proposed change resulting from a dispute.

Figure 9-2 DRR Status Values

9.2 CDRL TRACKING PROCEDURES

CDRLs define the type and frequency of required data items. CDRL tracking procedures help monitor development of the software and its associated documentation. The PMO and/or the Contracting Officer's Technical Representative (COTR) approves and tracks all data items required by the program/application.

The PMO needs to know when a newly generated document requires authentication or someone proposes a change to an existing authenticated document. The PMO schedules and monitors a full review of the new or changed document in accordance with DRR procedures. This review cycle may also be set by contract and is defined in the CDRL.

For documents needing revision, the PMO notifies the development contractor as prescribed by the contract. With coordination, CM can assure that all the necessary CDRL tracking data, including contract, distribution, deliverables, review schedules, library entries, and DRRs are entered into the CMDB. Once this information is included in this database, all interested Government and contractor personnel can access it.

The CM library maintains information on, and copies of, CDRL documents approved by the Government. The library distributes documents as requested by the PMO. Library services, the DRR process and CDRL tracking provide complete life cycle management for all program documentation.

10. ACTION ITEM PROCESS

AIs are a management tool used by the PMO and executive manager to identify and track issues for the program. AIs may be initiated at user group meetings, reviews, or other technical exchange meetings. They may be initiated by any participant (identified as the originator of the AI) in the software development process and are tracked by CM to solution/closure. The process is illustrated in Figure 10-1.

10.1 AI PROCESS DESCRIPTION

□ *Step 1* - The AI is initiated by a participant of the program at any time during the life cycle of the program. The assignee of the AI takes responsibility for completing the AI.

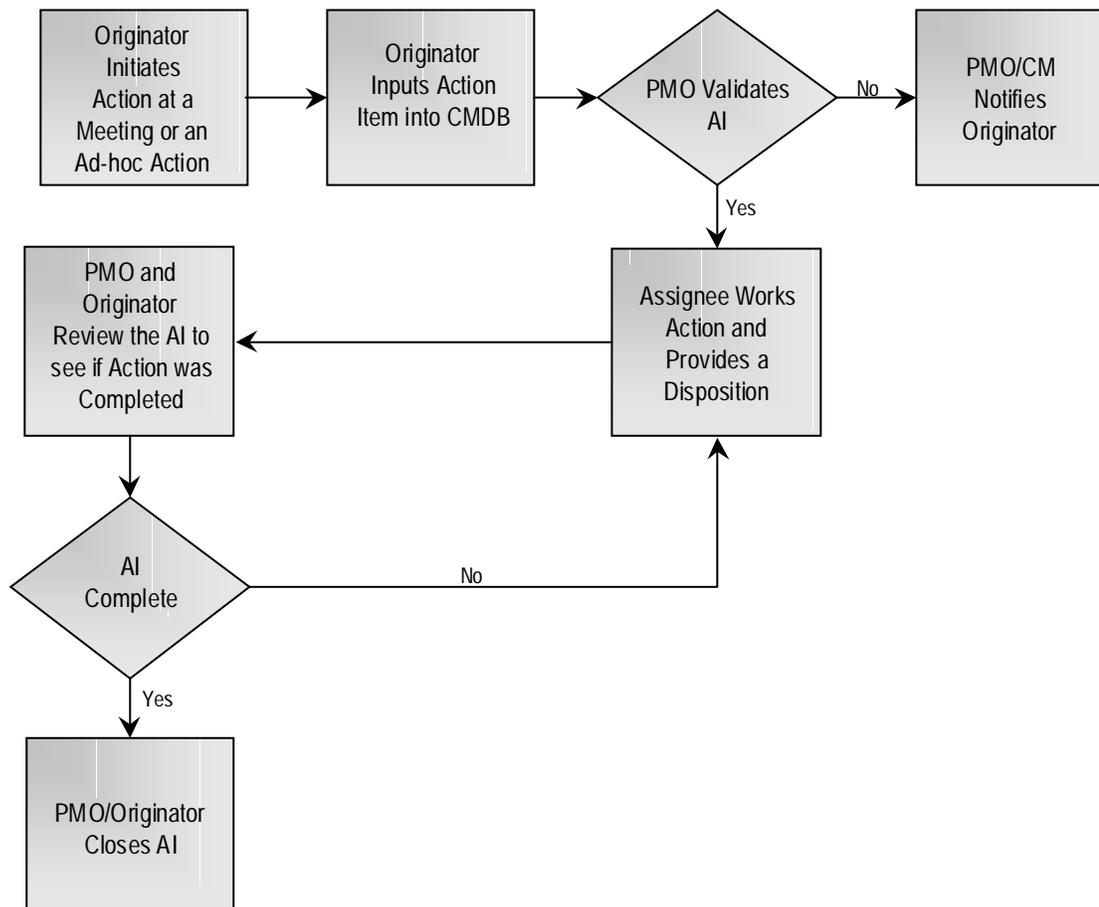


Figure 10-1 Action Item Process

- *Step 2* - The AI is entered into the CMDB.
- *For AIs generated outside of a PMO attended meeting:* The originator submits an AI through the Action Item (New Document) form of the CMDB. If the AI is classified, the originator must submit the AI through the Action Item (New Document) form of the CMDB on Intelink. Individuals who do not have Internet/Intelink access can use E-mail, message traffic via the AUTODIN communications network, facsimile, or regular mail.

All AIs will be sent to the PMO for screening and evaluation. CM will verify that all necessary information has been included in the form or message.

- If additional information is needed, the originator is contacted to provide further details or clarification.
- *Step 3* - Once the AI is validated, the CMDB will notify the originator and assignee via automatic notification.
- *Step 4* - The assignee will perform the action needed and update the status and disposition in the CMDB.
- Suspense dates on AIs are periodically checked. If an AI is overdue, notification will be sent to the PMO, originator, and assignee for review.
- *Step 5* - The PMO and originator will review the AI to determine if the action was performed satisfactorily.
- *Step 6* - If the AI was performed satisfactorily, the originator and PMO will close the AI. If not, Step 4 begins again.

10.1.1 Action Item Status Values

Each AI record has various states in its lifecycle. Figure 10-2 identifies the status values that can be assigned to a AI record from initial identification through closure.

Status Value	Role	Definition
Open	CM/PM, Originator, Assignee	The AI has been identified and documented.
New	CM/PM	A flag status, which indicates that the AI has not been reviewed by the PM. This is the initial status of a AI for a PM.
In Review	CM/PM	The PMO is reviewing the AI. The Action Item has not been validated as an actual AI.
Revised	Assignee, Originator	Indicates that the AI has been changed.
Still In Work	Assignee	Identifies that the AI is being worked.
Overdue Suspense Date	CM/PM	Identifies that the suspense date by which the action was to be completed has passed and resolution of the action is overdue.
Revised Suspense Date	CM/PM	Identifies that the suspense date by which the action was to be completed has changed.
Rejected	CM/PM	The AI was not accepted as a required action.
Closed	CM/PM, Originator, Assignee	The AI is complete.

Figure 10-2 Action Item Status Values

11. SOFTWARE RELEASES

A software release is the distribution of the latest baselined version of an application. Software release information is tracked in the CMDB for all supported programs. This allows the Government to track who has the software and any COTS software distributed with a release. The information is obtained from various sources, including but not limited to the SVD, PMO, e-mail from sites, and information obtained from the development contractor. The purpose of the Release section of the CMDB is to provide a consolidated location for information on software versions.

11.1 PMO RESPONSIBILITIES

The PMOs have the responsibility to:

- Provide a schedule for the release including anticipated test and distribution dates.
- Provide a list of sites and users that will receive the release.
- Provide a list of documents to be included in the release package, including identification of those documents distributed to the field.

11.2 SITE RESPONSIBILITIES

The sites have the responsibility to:

- Maintain a Profile/POC Information on the CMDB to ensure any changes to pertinent information are reflected in a timely manner.
 - POCs provide information used to identify focal points for applications that are responsible for an application at a given location. The information in POCs is used for preparing software release distribution packages, distributing PR/CR reports and updates, and assisting in the evaluation of program PRs/CRs. Up to date POC information is critical in ensuring the timely distribution and receipt of new software and program information.
- Notify CM if their site is leaving or joining the program software community.
- Provide in a timely manner a current mailing address for the distribution of the release when requested by CM.
- Return the receipt acknowledgment letter provided in each software and/or documentation release package(s) via mail, facsimile, or automated form on the CM homepage.
- Provide CM the installation date and any installation details immediately following installation of all software releases via Library Audit in the CMDB. Users who do not have Internet/Intelink access can use E-mail, phone, or message traffic.
- Provide estimated installation date and explanation for any delay if the site is not planning on installing within 30 days.

- Verify incorporation of PRs/CRs into the release and update the status of those PRs/CRs that are now fixed or are still valid in the CMDB.

11.3 CM RESPONSIBILITIES

CM has the responsibility to:

- Input site information obtained from a data call into the CMDB.
- Notify PMOs of changes to site profile information as required.
- Reproduce the software release and appropriate documentation.
- Distribute the software release and appropriate documentation to the PMO specified sites.
- Maintain the distribution lists on which sites received the software release in the CMDB.
- Upon receipt of the acknowledgment letter provided in each software and/or documentation release package(s), update the CMDB with the appropriate information.
- Coordinate user requests with PMOs.

11.4 DEVELOPMENT CONTRACTOR RESPONSIBILITIES

The development contractor has the responsibility to:

- Notify CM of any changes or updates to user/site information based upon data provided by site representatives, help desk activities, or other methods.
- Review CM user site information prior to distribution of any software release directly from the developer's site to ensure correct mailing addresses and POC information.
- Notify CM of distribution software releases made by the developer per contracted requirement. All information to maintain the release records can be submitted via softcopy or hardcopy.

11.5 EMERGENCY RELEASES

Emergency releases will be used to effect repairs to baselined versions or revisions that are not capable of operating due to Impact Code 1 problems as described in Section 8.2.3. These problems affect either the overall operational capability or severely compromise national security. The PMO and executive manager will coordinate emergency releases. Any changes will be fully documented and incorporated into the next scheduled release of the software. Patch numbers, distribution, and installation information must be reported to CM as soon as possible for tracking purposes and easy identification of those sites, which are above the current approved baseline.

12. ACRONYMS

AC2ISRC	Aerospace Command and Control Intelligence Surveillance and Reconnaissance Center
ADM	Acquisition Decision Memorandum
AFRL	Air Force Research Laboratory
AI	Action Item
AIG	Address Indicator Group
AUTODIN	Automatic Digital Network
CCB	Configuration Control Board
CCCB	CUBIC Configuration Control Board
CDRL	Contract Data Requirements List
CI	Configuration Item
CM	Configuration Management
CMDB	Configuration Management Database
COTR	Contracting Officer's Technical Representative
COTS	Commercial Off The Shelf
CSCI	Configuration Software Control Item
CR	Change Request
CUBIC	Common User Baseline for the Intelligence Community
DAWS	Defense Automated Warning System
DBDD	Data Base Design Document
DD	Defense Document
DEXA	DODIIS Executive Agent
DIA	Defense Intelligence Agency
DID	Data Item Description
DMB	DoDIIS Management Board
DMS	Defense Messaging System
DOD	Department of Defense
DODIIS	Department of Defense Intelligence Information System
DRR	Document Review Report

DSN	Defense Security Network
ECP	Engineering Change Proposal
EIA	Electronic Industries Association
ESC	Electronic Systems Center
GOTS	Government Off The Shelf
ICD	Interface Control Document
ICWG	Interface Control Working Group
ID	Identification
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IESS	Imagery Exploitation Support System
IFEB	Integration and Interoperability Branch
IMA	Intelligence Mission Application
IPAT	In-plant Acceptance Test
ISO	International Organization for Standardization
IV&V	Independent Verification & Validation
JITC	Joint Interoperability Test Command
JITF	Joint Integration Test Facility
JTPM	Joint Test Planning Meeting
JTFF	Joint Test Readiness Review
MAP	Multiple Application Problem
MAXI	Modular Architecture for Exchange of Information
MDA	Milestone Decision Authority
PCMR	PR/CR/MAP/Requirement
p ³ I	Pre-Planned Product Improvement
PLA	Plain Language Address
PM	Program Manager
PMO	Program Management Office
POC	Point of Contact
PR	Problem Report

QA	Quality Assurance
SCI	Sensitive Compartmented Information
SETA	System Engineering and Technical Assistance
SIMO	System Integration Management Office
SPR	Software Problem Report
SVD	Software Version Description
TEMS	Technical and Engineering Management Support
TF	Test Finding
U&S	Unified and Specified
WP	Workplan