

**PRETRIAL SERVICES AGENCY
FOR THE DISTRICT OF COLUMBIA**



**CONFIGURATION MANAGEMENT
PLAN**

January 24, 2012

Rev 1.2

Version	Change Comments	Date	Author
1.0	Initial Draft	1/4/2005	D. Swingle
1.1	Updates	6/1/2005	D. Caravantes
1.2	Review and Updates	1/24/2012	D. Caravantes

Table of Contents

INTRODUCTION	3
1.1. PURPOSE	3
1.2. BACKGROUND	3
1.3. SCOPE	3
2. CONFIGURATION MANAGEMENT	4
2.1. CONFIGURATION MANAGEMENT DEFINED	4
2.2. CONFIGURATION MANAGEMENT PLAN OVERVIEW	4
2.2.1. <i>Necessity of a Configuration Management Plan</i>	4
2.2.2. <i>When to Develop a Configuration Management Plan</i>	5
3. CONFIGURATION MANAGEMENT PLAN COMPONENTS.....	6
3.1. ROLES AND RESPONSIBILITIES	6
<i>OIT Director</i>	6
<i>Development Team</i>	6
<i>Software Architect</i>	7
<i>System Users</i>	7
3.2. COMMUNICATIONS.....	7
3.3. BASELINE CONFIGURATION	7
3.4. CONFIGURATION CHANGE CONTROL	7
3.5. SECURITY CONSIDERATIONS.....	7
4. BASELINE CONFIGURATION	8
4.1. BASELINE INTRODUCTION.....	8
4.2. DEFINING THE BASELINE	8
4.3. DOCUMENTING THE BASELINE	8
4.4. SECURING THE BASELINE.....	8
5. CONFIGURATION CHANGE CONTROL.....	9
5.1. CHANGE CONTROL INTRODUCTION	9
5.2. THE CHANGE REQUEST.....	11
5.3. CM RECEIVES CHANGE REQUEST	12
5.4. OIT RECEIVES CHANGE REQUEST	12
5.5. CMCB EVALUATES RECOMMENDED CHANGE PACKAGE	13
5.6. OIT IMPLEMENTS CHANGE PACKAGE	13
5.6.1. <i>Implementation</i>	13
5.6.2. <i>Prepare/Integrate Release</i>	13
5.6.3. <i>Update Documentation</i>	13
5.6.4. <i>CM Closes Out Change Request</i>	14
5.7. EMERGENCY OR MINOR CHANGES.....	14
5.8. CHANGE CONTROL TASK LIST.....	15
6. SECURITY CONSIDERATIONS.....	16

6.1.	ACCESS RESTRICTIONS.....	16
6.2.	CONFIGURATION SETTINGS	16
6.3.	CONTINUOUS MONITORING	16
7.	APPENDIX A – CHANGE REQUEST FORM.....	17
8.	APPENDIX B – CHANGE CONTROL CHECKLIST	18
9.	APPENDIX C – DETERMINE SECURITY IMPACT OF CHANGE	20
10.	APPENDIX D - CHANGE PACKAGE TEMPLATE.....	21
11.	APPENDIX E - CHANGE CONTROL PROCESS - TASK VIEW	23
12.	APPENDIX F - CONFIGURATION CHANGE CONTROL TASK LIST	24
13.	APPENDIX G – DETERMINE MAGNITUDE OF CHANGE REQUEST	25

INTRODUCTION

1.1. Purpose

The material and procedures in this Configuration Management Plan (CMP) provide guidance an in-depth and consistent approach to configuration management for the District of Columbia Pretrial Services Agency (PSA). The guidance is to be used by individuals responsible for, or involved in the configuration management of PSA's business and mission critical systems. Adherence to the framework and procedures will ensure compliance with PSA's Configuration Management Policy and Management Instructions and FISMA regulation.

1.2. Background

Configuration Management is the process of establishing and maintaining the technical integrity of a system throughout its life cycle by systematically identifying, controlling, and accounting for all changes made to a system. According to the *PSA Configuration Management Policy*, a Configuration Management process shall be developed for each business and mission critical system to effectively manage and track system changes. As part of the documentation required for a system to be certified and accredited, the agency's Configuration Management procedures require a CMP as one of the security documents that must be developed. This document provides a structured method of recording the Configuration Management process for a system in a CMP.

1.3. Scope

This document outlines the major elements of a CMP and provides a description of the content for each element. In addition, it presents an overview introduction to Configuration Management and CMPs, including what Configuration Management is, what a CMP is, and why and when a CMP should be developed. The configuration change control process, which is the most important element of the CMP, will be discussed in depth.

This handbook is based on the *Pretrial Services Agency Security Policy*; Office of Management and Budget (OMB) Circular A-130, Appendix III, *Security of Federal Automated Information Resources*; and other applicable federal information technology (IT) security laws and regulations.

2. CONFIGURATION MANAGEMENT

2.1. Configuration Management Defined

Configuration Management is the systematic identification, documentation, and control of system elements by recording and reporting change processing and the status of system implementation. These activities assist in verifying compliance with specified system requirements as well as establishing and maintaining the technical integrity of a system throughout its life cycle. The successful implementation of Configuration Management activities results in an established and documented system baseline, effective management and tracking of changes made to a system and related documentation (version control), and effective risk management.

2.2. Configuration Management Plan Overview

A Configuration Management Plan (CMP) is a living document that identifies Configuration Management roles, responsibilities, resources, and formal processes and procedures to ensure that all proposed changes to a system are evaluated and approved before implementation. A CMP is essential for effective Configuration Management as it relates to activities such as software versioning, system-wide upgrades, replacements, and deployments. In addition, it is a key process to assure a system is secure.

A CMP should include and address Configuration Management roles and responsibilities, channels of communication, system configuration baseline, configuration control process, and Configuration Management resources. Additional elements of the CMP document itself should also include a cover page, table of contents, an executive summary, and an introduction section. A detailed description of each CMP component is provided in Section 3.

2.2.1. *Necessity of a Configuration Management Plan*

Federal regulation and guidance, and the agency's Configuration Management Policy and Management Instruction, a Configuration Management process must be developed and documented in a CMP for all system's. Developing a CMP is critical for implementing Configuration Management and ensures the following:

Changes to the configuration are identified and evaluated to determine the impact to system security before implementation.

Any changes made to the system are documented and tracked. Ideally, this process begins at the system development stage and is carried out until the system is replaced. Because each change is tracked from initial system development through completion, a thorough history of changes is created for that system. For example, it may become necessary to update source code for an application due to system enhancements. The resulting changes will alter the application's configuration. Also, the need for a version upgrade is a configuration change that must be thoroughly analyzed, since it can affect security, system performance, and functionality. This change should be documented in a formal process to provide a historical representation of one of the changes occurring throughout the system development life cycle (SDLC).

Configuration is documented ensuring that version control is maintained. Upgrades and additions are easily implemented and tracked because of hardware and software controls in a formal Configuration Management process.

The system documentation includes information on the system specification and configuration design, ensuring that, as part of the CMP, system documentation can be checked to verify whether the designed configuration allows the system to achieve its objective. For example, if a system's design needs to be modified to implement a stringent password for users, it could be determined that this change will be included in a later version. Prior to the new version's implementation, all changes should be documented to ensure that version control is maintained.

The configuration is verified and tested against the initial baseline to ensure that all changes have been maintained and documented for any future parties involved in the Configuration Management process.

Inventory on the system, including the manufacturer, model type, and software version, is recorded and tracked, ensuring access to the most recent system information.

2.2.2. When to Develop a Configuration Management Plan

As a rule, a CMP should be developed at the beginning of the SDLC to ensure that a Configuration Management process is initiated to control, track, and maintain all changes that are made to the system throughout the SDLC. A CMP document should be reviewed and updated as needed throughout the entire SDLC. It is possible to develop a CMP after the system has been deployed; however, existing system documentation (i.e., system security plan, system configuration documents, system maintenance records, vendor manuals, system configuration diagrams, and security-related information) is more heavily emphasized in the development of the CMP.

3. CONFIGURATION MANAGEMENT PLAN COMPONENTS

3.1. Roles and Responsibilities

The following table describes the roles and responsibilities that are part of configuration management.

Role	Responsibilities
<i>PSA Office of Information Technology (OIT)</i>	OIT is responsible for implementing and testing systems to assure that they meet specified functional requirements. The OIT has several responsibilities in the change control process, including: <ul style="list-style-type: none"> • Analysis of change request • Preparation/Evaluation of change package (CP) • Implementation/Integration of CR • Updating necessary documentation with respect to the new change.
<i>OIT Director</i>	Responsible for setting forth Management Instructions regarding Configuration Management and implementing CM at the highest level for the OIT.
<i>Configuration Manager (CM)</i>	The CM is “middle man” throughout the change control process. He/she will receive change requests or change packages, evaluate them for completeness and send them on through the process or reject them. Other responsibilities for the CM include: <ul style="list-style-type: none"> • Implement the configuration management plan • Coordinate meetings of the CMCB • Maintaining consistent records of system changes and their statuses • Maintaining an accurate system baseline.
<i>Change Management Control Board (CMCB)</i>	The CMCB is the governing body for CM policy and guidance. The responsibilities of the CMCB include: <ul style="list-style-type: none"> • Evaluate and approve change requests. • Meet on a regular basis (at least once per month) to discuss relevant issues and deferred change requests. • Review and update CMP as necessary. • Ensure life-cycle cost savings • Ensure that proposed changes do not adversely affect additional systems, established priorities, and budgets.
<i>System Development Director</i>	The Director of System Development or other designated individual serves as the authority for all matters of Configuration Management for the business or mission critical system. The System Development Director is responsible for developing functional requirements and verifying that the requirements are implemented appropriately. This individual may also play a role in establishing the Configuration Control Review Board (CCRB) and may be involved in the selection of the CCRB members.
<i>Development Team</i>	The System Development team is largely responsible for the success or failure of a CMP. The team is ultimately in charge to design, develop, and test any changes

Role	Responsibilities
	made to an application. In addition, team members are responsible for tracking their work (source code, etc) and reporting changes to the Configuration Manager and Software Architect for baseline so that all work can be documented and tracked.
<i>Software Architect</i>	Technical lead of the System Development team. Responsible for making sure the team is following through with their responsibilities with regards to implementing functional requirements and configuration management tasks. The Architect is also responsible for maintaining the system baseline.
<i>System Security Officer (SSO)</i>	Responsible for assuring that all Configuration Management activity is consistent with the requirements of his/her respective system security plan (SSP), all security policies and management instructions with applicable coverage.
<i>System Users</i>	Can initiate a change request for a given system.

3.2. Communications

Channels of communication will be used to share information regarding Configuration Management (e.g., upgrades, application changes, technical notices, and version control). It is important that all necessary parties have access to information so that decisions can be made with clear and concise knowledge, and with the most recent data available. However, access restrictions must be observed throughout communication channels (see 6.1 – Access Restrictions).

3.3. Baseline Configuration

A baseline is defined as an authorized software work product that can only be changed through formal change control procedures. The baseline configuration is further discussed in Section 4.

3.4. Configuration Change Control

The Configuration Change Control sets forth the structure and process by which a system can be changed. It will include the channels of approval and the process that a change request must go through before it can be implemented. Change control is further discussed in Section 5.

3.5. Security Considerations

It is important to maintain secure practices throughout configuration management. Without secure practices, systems are vulnerable to attacks and availability problems. Security should be woven into baseline configuration and the configuration change control process, but there are three other areas where secure practices are required:

- Access Restrictions
- Configuration Settings
- Continuous Monitoring

4. BASELINE CONFIGURATION

4.1. Baseline Introduction

The term baseline is used to refer to a reference point by which requirements, design and changes to requirements and design can be measured. Essentially, a baseline is an approved snapshot of system at a given time during the SDLC. Well-defined baselines will be a meaningful starting point for defining future changes.

4.2. Defining the Baseline

Baselines are defined at various points in time. During the development phase of the SDLC, a baseline may be defined with each development milestone. Once development is completed, the finished product with regards to requirements will be the baseline by which all future changes are measured. Each time a new version of the product is finalized, a new baseline will be established.

4.3. Documenting the Baseline

It is imperative that each baseline is documented so that the baselines can be tracked in the event that it becomes necessary to fall back to a previous version of the system. At the minimum, the following data should be tracked when a baseline is finalized:

- Date/time of the baseline snapshot
- Description of the baseline (i.e. – Initial system release or description of changes incorporated since the last baseline)
- List of files/source code that are associated with the baseline should include:
 - File Name
 - Last updated date/time
 - Description of change to file
- Inventory of information system physical components:
 - Manufacturer
 - Type
 - Serial number
 - Version number
 - Location (i.e., physical location and logical position within the information system architecture)

4.4. Securing the Baseline

Automated mechanisms should be used to maintain the baseline configuration. The use of a tool such as Microsoft Visual SourceSafe is a reliable way to track source code and system documentation. Access to the system baseline should be strictly enforced, and limited only to those individuals that require access. In addition, the baseline should be backed up daily, and the backup media stored off-site.

5. CONFIGURATION CHANGE CONTROL

5.1. Change Control Introduction

Configuration change control represents the vital process by which a business or mission critical system undergoes changes, and how those changes are initiated, approved and recorded. Configuration and change control involve these parties:

- Configuration Change Manager (CM)
- PSA Office of Information Technology (OIT)
- Configuration Management Control Board (CMCB)

Configuration and change control involves the following activities:

1. Receive Change Request (CR)
2. Analyze CRs
3. Prepare/evaluate change packages (CPs)
4. Approve CPs
5. Implement approved CPs
6. Prepare releasable products
7. Integrate and release the product suite
8. Maintain consistent documentation throughout the process.

Figure 5.1 represents a model for Configuration Change Control Process. It is important to verify that each step in the flow is accompanied by verification that security related tasks are being performed throughout the process. For a checklist of these tasks and their associated steps in the change control process, see Appendix B.

Configuration Change Control Process Flow

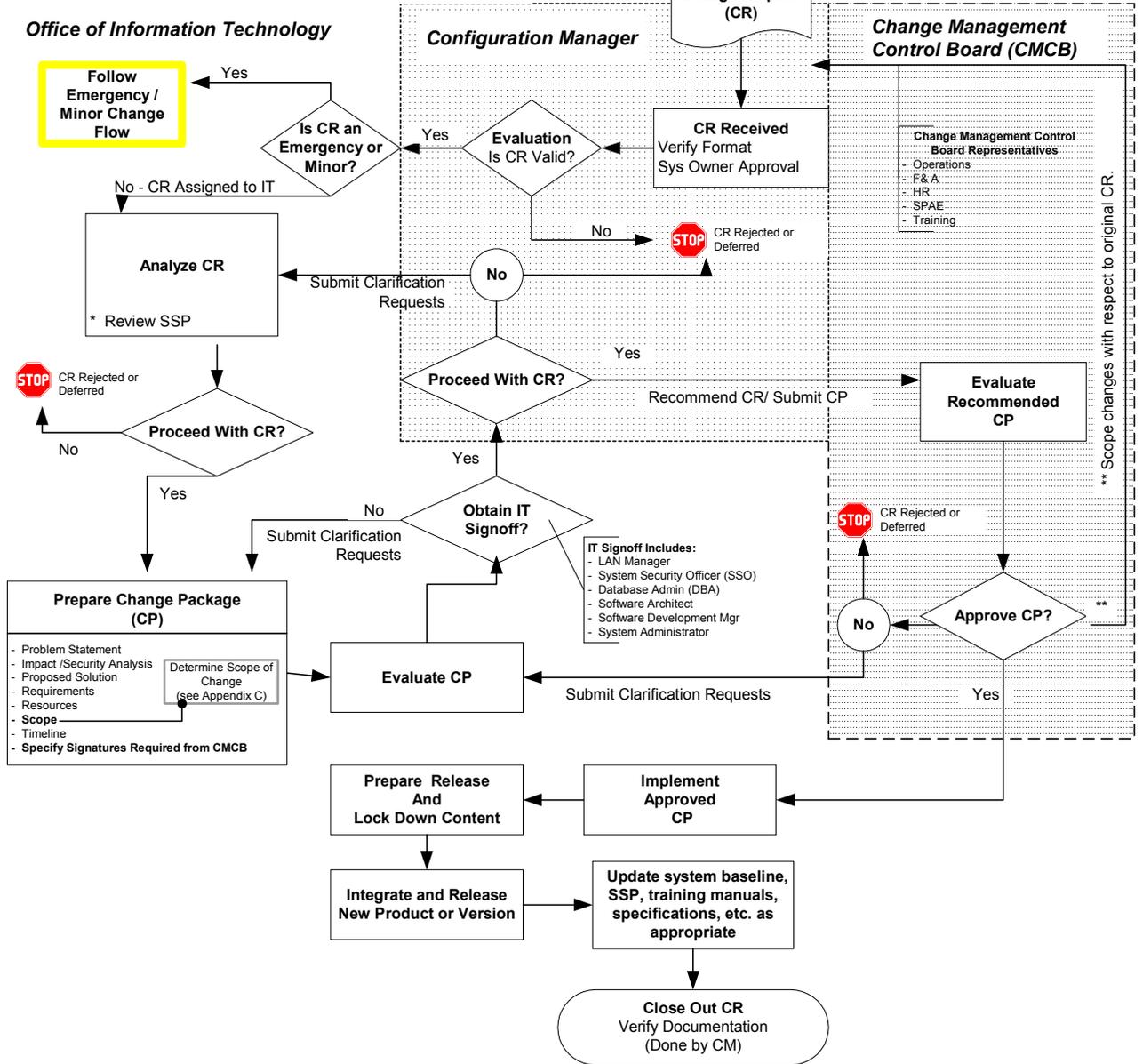


Figure 5.1 – Configuration Change Control Process

5.2. The Change Request

As figure 5.1 indicates, the change control process is initiated by a change request (CR). A CR can be initiated at any level throughout the agency and normally falls into one of four categories:

- New feature/function
- Change to existing feature/function
- Software bug
- Security issue

The initiation of a change request is illustrated in Figure 5.2 below.

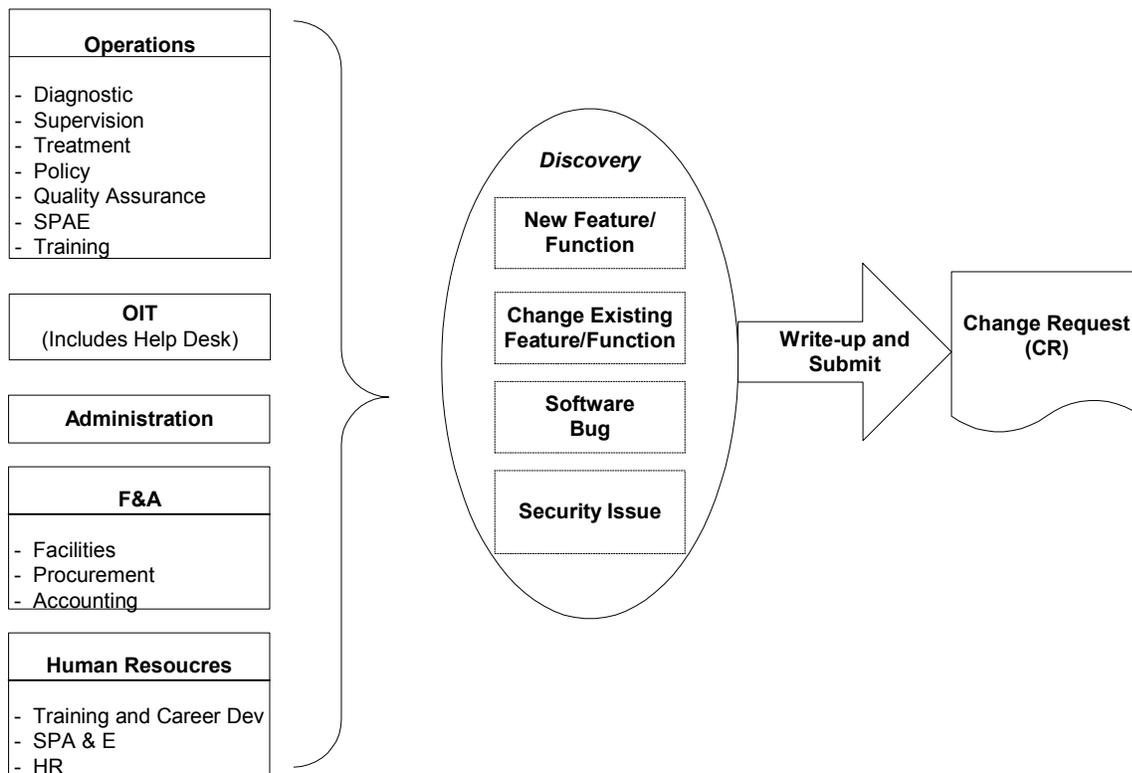


Figure 5.2 – Initiating a Change Request

5.3. CM Receives Change Request

When the CM receives a change request, the CR will proceed through the flow described in Figure 5.1. The first thing the CM will do when receiving a new change request is record it with a number, description, date-received and status. The CM will then screen the CR for completeness and determine the magnitude of the change (see Appendix G – Determine Change Magnitude). It may be necessary for the CM to consult with OIT in order to determine the change magnitude. Once the change magnitude has been determined, the CM will send the CR on for analysis and approval by OIT.

If the CM determines that a change request is invalid, they have the authority to reject a CR at this point, or defer it. The table below describes what occurs in each case:

CR Status	Result
CR is rejected	<ul style="list-style-type: none"> ○ CR status updated to “Rejected” ○ CR originator notified
CR is deferred	<ul style="list-style-type: none"> ○ CR status updated to “Deferred” ○ CR originator notified ○ CMCB will review deferred CRs at their regularly scheduled meetings to determine if they can be re-initiated.

Figure 5.3 – Change Request is Rejected or Deferred

5.4. OIT Receives Change Request

1. When a CR is received by OIT, it will undergo the following steps:
 1. If OIT determines that the change is neither an emergency nor minor change, they will further analyze the CR and determine if the CR should be carried out. Technical validity, technical merit and technical impact of the CR will play a role in the decision. OIT can also, at this point, notify the CM and either reject or defer the CR (see Figure 5.3 – CR is Rejected or Deferred).
 2. System Security Officer reviews the CR
 - a. Review and apply diagram in Appendix C to change request to determine security impact of the proposed change.
 3. After the decision is made to proceed with the CR, OIT will write the change package (CP). For a template of the CP, see Appendix D. Elements of the CP include:
 - a. Problem Statement
 - b. Impact/Security Analysis – should include additional minor changes to the software that may be necessary to implement the CR.
 - c. Proposed Solution
 - d. Requirements needed
 - e. Resources to carry out CR
 - f. Timeline
 - g. Determine if CR should enforce a software version/release change.
 - h. Scope – used to determine signatures that will be required from the CMCB. See Figure 5.4 – Determine CR Scope.

4. Once the CP has been assembled, it must obtain signoff from the necessary personnel within the OIT. Those individuals include:
 - a. System Development Director
 - b. System Security Officer (SSO)
 - c. Database Administrator
 - d. Software Architect
 - e. LAN Manager
 - f. System Administrator

5. Once IT signoff is obtained, the CP is sent to the CM, who will review it, update its status and send it on to the CMCB for approval. The CM can also opt to send it back to the OIT for clarification, reject the CP or defer the CP (see Figure 5.3 – CR is Rejected or Deferred).

5.5. CMCB Evaluates Recommended Change Package

The CP will be evaluated by the CMCB. At this point, the CMCB has three options: they can approve the CP (based on signatures that the CP specifies) and send it to OIT for implementation, they can reject or defer the CP, or they may choose to resubmit the CR in light of new requirements that the CP does not fully cover (in which case, the new CR will be sent back to the beginning of the flow).

5.6. OIT Implements Change Package

Following sign-off from the required members of the CMCB, OIT has four duties: implement the CR, prepare the release of the CR, release the new product, and update the necessary documentation.

5.6.1. Implementation

Implementation involves the software development and testing of the CR. During implementation, the OIT may have to initiate additional minor changes in the software. These changes should have been incorporated into the CP, but may have gone undetected at the time the CP was prepared. If that was the case, the changes can still be made, but the CM must be notified of the additional changes once the new product has been fully tested.

5.6.2. Prepare/Integrate Release

Once the changes have been implemented and tested, the new product is ready for release. OIT will lock down content to prepare for the release, and verify that all updated changes have been relayed to, and recorded by the CM. Next, the new product will be released to the user community, and that the CM notified to change the status of the CR to “Complete”. The CM must also record any version upgrade that has taken place as a result of the CR.

5.6.3. Update Documentation

Following product release, it is necessary to update all documentation associated with the change. Documentation will include:

- Product baseline
- System Security Plan (SSP)
- Training materials
- Additional documentation as needed

5.6.4. CM Closes Out Change Request

The final step in the change control process is for the CM to close out the CR. In addition to finalizing the status, the CM reviews the CR to make sure that it is organized, and that all appropriate documentation is accounted for. The CM is responsible for maintaining an accurate history of all CR's.

5.7. Emergency or Minor Changes

As previously mentioned, there are occasions when a CR will have to be expedited through the change control process. In the case of emergency or minor changes, it will not be necessary for a CR to follow the complete process described in flow from Figure 5.1. Rather, in an effort to expedite minor and emergency changes, the change control process will be represented by the figure below:

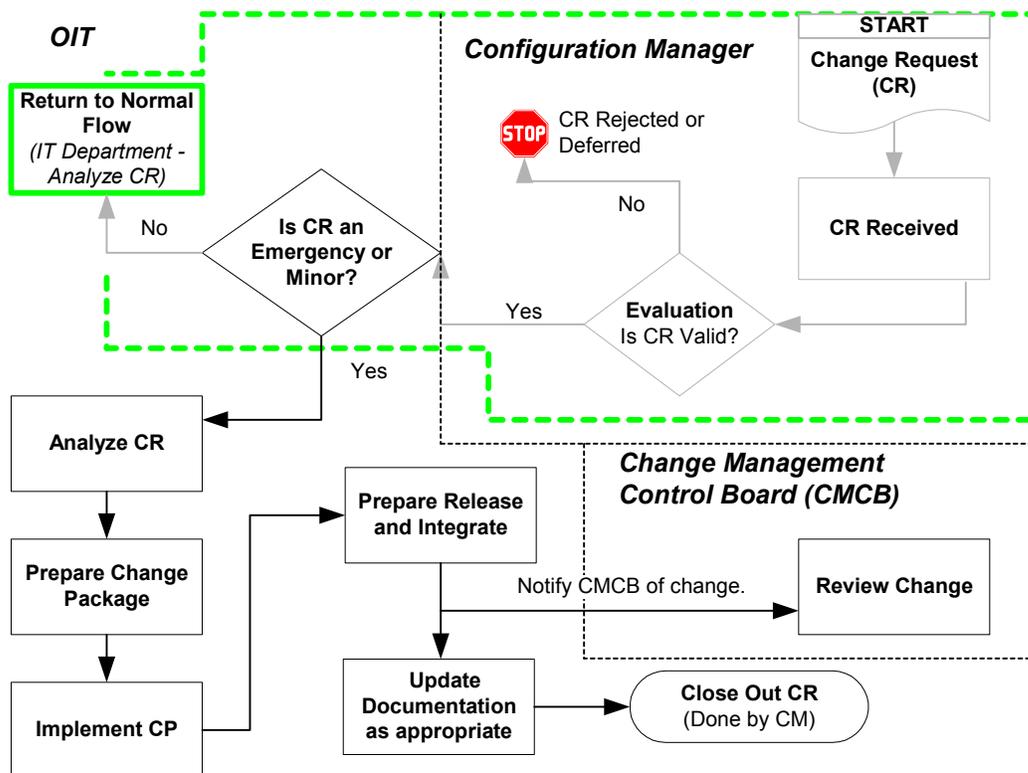


Figure 5.4 –Control Flow for Emergency or Minor Changes

Figure 5.4 reflects the necessity of the change control process to be flexible enough to be efficient when changes are needed fast. As such, it is not required that the CM or CMCB approve change packages for minor or emergency changes, but rather OIT is entrusted to follow through with minor or emergency change requests almost in their entirety. There are two main reasons for this. One is that, for emergency changes, the time that is required for a CR to proceed through the entire process flow could create security or system availability issues that will remain open until the CR is implemented. Another is simply that, for minor changes, the level of involvement or input from the CM and CMCB is not necessary.

5.8. Change Control Task List

The configuration change control process flow can be summarized by a list of tasks. For a reference to how identified tasks map to the flow chart, see Appendix E (Configuration Change Control Process Flow - Task View), and Appendix F (Configuration Change Control Task List).

6. SECURITY CONSIDERATIONS

In addition to maintaining a well-documented system baseline and following a consistent and organized change management process, information and process assurance is required. For this, there are certain security considerations that must be incorporated

6.1. Access Restrictions

To limit access to the implementation and review of system changes, the agency will enforce access restrictions based on a limited number of personnel respective of their job function. To this end, the system administrator will distribute unique user ids and passwords to access the system servers, file shares, and development environment tools, so to prohibit unauthorized access and to facilitate the accurate tracking of the individuals making changes. Separation of duties requirements as prescribed in the Access Control Policy and Management Instruction, should be practiced as resources allow.

6.2. Configuration Settings

The Agency configures the default security settings of information technology products to the most restrictive mode consistent with information system operational requirements, and also configures the information system to provide only essential capabilities while specifically prohibiting the use of the unnecessary ports, protocols, and/or services. In addition, the agency will continuously monitor the security settings to determine if the current configuration fulfills the security requirements.

6.3. Continuous Monitoring

The monitoring of configuration changes is accomplished throughout the change control process flow. The CM plays the most critical role in assuring that the agency maintains an accurate and thorough record of system baselines and changes.

In addition to maintaining documentation, it is essential that the agency continuously monitors system changes and conducts impact analysis of changes regularly. Changes should be monitored with respect to system performance, efficiency and user productivity. The continuous monitoring of configuration changes can help identify costs linked to failures such as replacement of equipment and user productivity.

Finally, the monitoring of configuration changes also includes an audit of system programmer activities including use of system utilities. These records help to link changes to the individual responsible for implementing them.

7. APPENDIX A – CHANGE REQUEST FORM

Change Request Form		
CR Originator:	Priority:	CR Tracking Number
System Owner Signature:	<input type="checkbox"/> Critical <input type="checkbox"/> Routine <input type="checkbox"/> Administrative	
Date:	Title of Change:	
Description of Change:		
		<input type="checkbox"/> Continues on attached page
Product Identification Impact, Software		<input type="checkbox"/> Continues on attached page
Product Identification Impact, Hardware		<input type="checkbox"/> Continues on attached page
Product Identification Impact, Documentation		<input type="checkbox"/> Continues on attached page
Security Impact:		
Business Impact:		
		<input type="checkbox"/> Continues on attached page
Justification of Change and Potential Impact if Change is Not Made:		
		<input type="checkbox"/> Continues on attached page
Estimated Number of Staff/Total Hours Needed:		Sites Affected:
Staff	Total Hours	
Comments:		

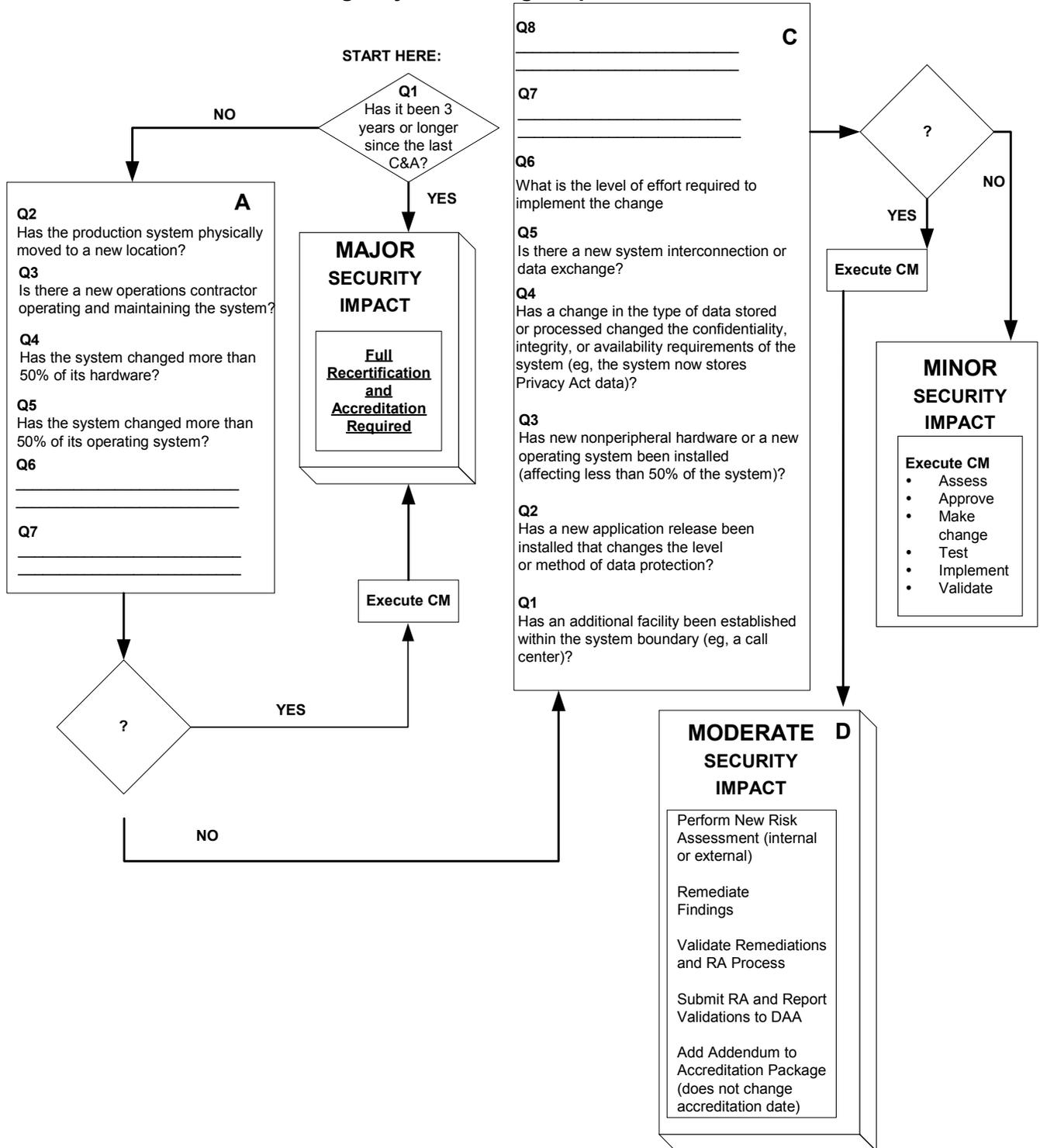
8. APPENDIX B – CHANGE CONTROL CHECKLIST

Configuration Change Flow Step	Security Task to be Performed	Performed By
Change Request Received	Document change request:	Configuration Manager
	Change Request Number	
	Date/Time	
	Description	
	Status	
Change Request Analyzed		OIT
	Identify any existing requirements in the baseline that conflict with the proposed change.	
	Identify any other pending requirement changes that conflict with the proposed change.	
	What are the consequences of not making the change?	
	What are possible adverse side effects or other risks of making the proposed change?	
	Will the proposed change adversely affect performance requirements or other quality attributes?	
	Is the proposed change feasible within known technical constraints and current staff skills?	
	Will the proposed change place unacceptable demands on any computer resources required for the development, test, or operating environments?	
	Must any tools be acquired to implement and test the change?	
	Will prototyping or other user input be required to verify the proposed change?	
	How much effort that has already been invested in the project will be lost if this change is accepted?	
	Identify any user interface changes, additions, or deletions required.	
	Identify any changes, additions, or deletions required in reports, databases, or data files.	
	Identify the design components that must be created, modified, or deleted.	
	Identify hardware components that must be added, altered, or deleted.	
	Identify the source code files that must be created, modified, or deleted.	

	Identify existing unit, integration, system, and acceptance test cases that must be modified or deleted.	
	Identify any help screens, user manuals, training materials, or other documentation that must be created or modified.	
	Identify any impact the proposed change will have on the project's software project management plan, software quality assurance plan, software configuration management plan, or other plans.	
	Quantify any effects the proposed change will have on budgets of scarce resources, such as memory, processing power, network bandwidth, real-time schedule.	
	Investigate all sources from which input can enter the program such as GUI, network reads, etc.	
Implement Change Request		OIT
	Are validation controls in place to ensure that necessary input data is correct and appropriate?	
	Are controls in place to capture system errors and display appropriate error messages?	
	Do software libraries reside in secure location and only accessed by personnel with the authority to make changes?	
	Are code reviews conducted before change is fully tested?	
Prepare Release		OIT
	Do software libraries reside in secure location and only accessed by personnel with the authority to make changes?	
Integrate and Release Product		OIT
	Do software libraries reside in secure location and only accessed by personnel with the authority to make changes?	
	Are server configuration settings appropriate to accommodate the change?	
	Document change request:	Configuration Manager
	Change Request Number	
	Date/Time	
	Description	
	Status	

9. APPENDIX C – DETERMINE SECURITY IMPACT OF CHANGE

Decision Process for Assessing a System Change--Operational Phase



10. APPENDIX D - CHANGE PACKAGE TEMPLATE



**District of Columbia Pretrial Services Agency
Application Change Control**

Change Package

Title:	Date:
CR Tracking Number:	System Owner:

1 - Problem Statement

2 - Impact/Security Analysis

3 - Proposed Solution

4 - Requirements

5 - Resources

6 - Change Scope

7 - Estimated Timeline

8 – Required Signatures from Change Management Control Board

9 – OIT Approval Signatures

System Development Director: _____
Signature Date

System Security Officer: _____
Signature Date

Software Architect: _____
Signature Date

Database Administrator: _____
Signature Date

LAN Manager: _____
Signature Date

System Administrator: _____
Signature Date

10 – Change Management Control Board Approval Signatures

Approval 1:
(Name/Title) _____
Signature Date

Approval 2:
(if necessary) _____
Signature Date

11 – Post-Approval Signatures (OIT)

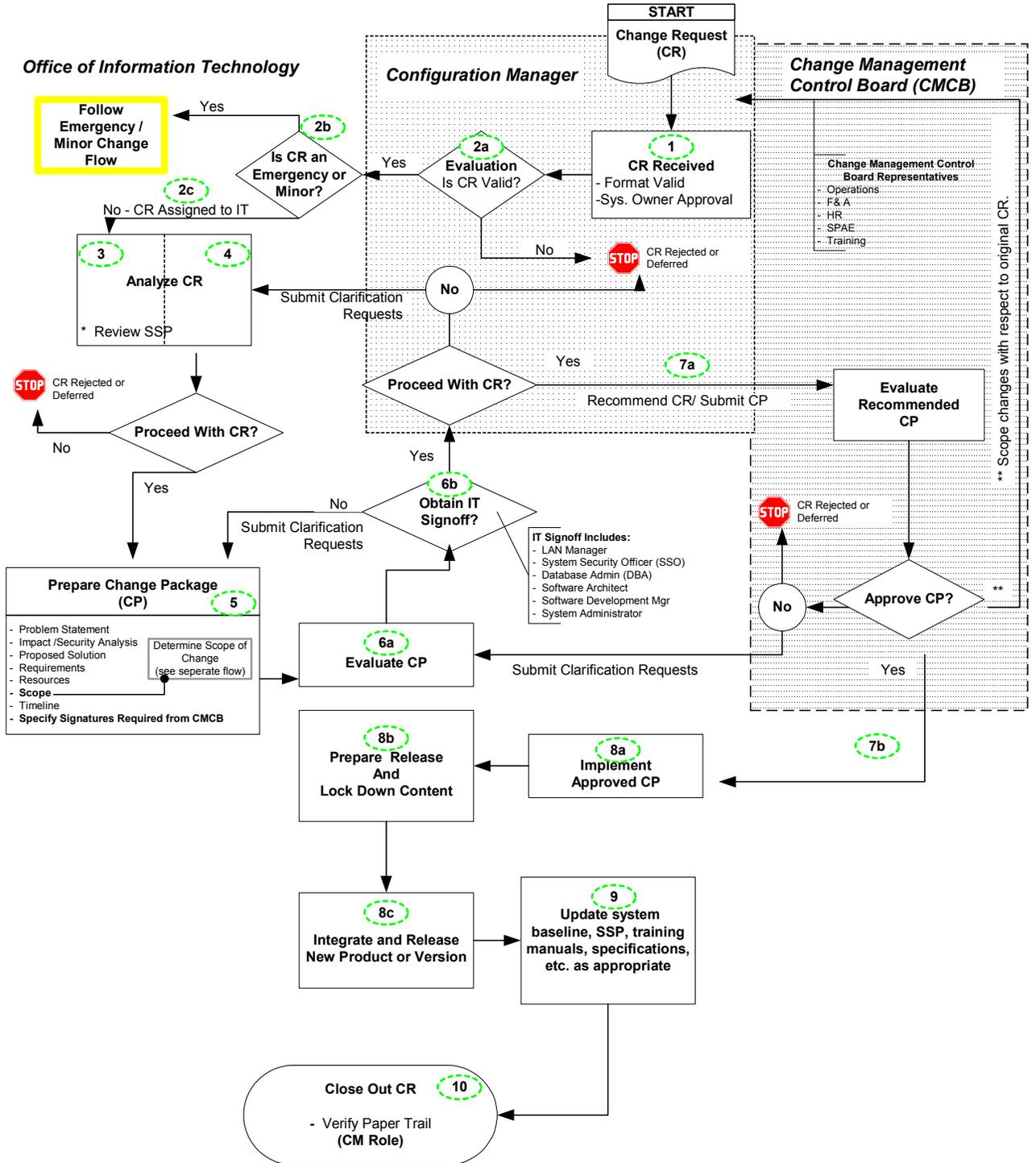
OIT certifies that this change request has been implemented and tested.

Name (Print) Signature Date

Comments:

11. APPENDIX E - CHANGE CONTROL PROCESS - TASK VIEW

Configuration Change Control Process Flow - Task View



12. APPENDIX F - CONFIGURATION CHANGE CONTROL TASK LIST

Configuration Change Control Task List				
Task #	Task Name	Related Subtasks	Responsible Role	Comments
1	CR Received	N/A	Configuration Manager	
2	Evaluation	2a) Is CR valid? 2b) Is CR Emergency or Minor? 2c) Assign CR to OIT	Configuration Manager	Must verify that 1 - CR is in valid format, and 2 - that relevant System Owner has approved CR.
3	Analyze CR	N/A	OIT	
4	Analyze CR	N/A	System Security Officer	
5	Prepare Change Package		OIT	
6	OIT Evaluation	6a) Evaluate 6b) Obtain IT Signoff	OIT OIT - LAN Manager OIT - SSO OIT - DBA OIT - Software Arch. OIT - Software Dev Mgr OIT - System Admin	The OIT Admin will have the responsibility to obtain signatures.
7	Guide CR through CMCB approval	7a) Send CR on to CMCB 7b) Give "Green Light" for OIT to implement	Configuration Manager	Specified signatures from CMCB must be obtained before OIT can commence implementation*
8	Implementation	8a) Implement CR (coding) 8b) Prepare Release 8c) Release New Product/Version	OIT	
9	Update Documentation		OIT	System Baseline SSP Training Manual(s) Software Specifications
10	Close out CR		Configuration Manager	Verify that all tasks are completed.

*CMCB signatures include:

Operations – Director of Operations, Deputy Director of Operations, Training and Career Development Director, SPA&E Senior Analyst

F&A – F&A Director

Human Resources – Human Resources Director

Lab – Lab Director/Technical, Lab Operations Director

New Funding – ITIMC Chair, System Development Director

13. APPENDIX G – DETERMINE MAGNITUDE OF CHANGE REQUEST

