1 of 1

Federal Emergency Management Information System (FEMIS) — System Platform

SOFTWARE CONFIGURATION MANAGEMENT PLAN (SCMP)

Version 1.1

F. E. Strycker, Jr.

July 19, 1993

**Prepared for the Federal Emergency Management Agency
under a related services agreement with the
U. S. Department of Energy
Contract DE-AC06-76RLO 1830**

**Pacific Northwest Laboratory
Richland, Washington 99352**

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

376

AUTHORIZATION:

Prepared by:

F.E. Strycker, Jr.
Project Support Manager

Approved by:

T. F. Coonelly, Jr.
Project Manager

Approved by:

C. D. Lee
Manager
Integrated Information Systems Group

Concurrence:

R. L. Cheatham
Manager
Computer Sciences Department

REVISION CHART:

	Revision Number	Approved

Contents

1.0 Introduction	1
1.1 Purpose	1
1.2 Scope	1
1.3 Definitions and Acronyms	1
1.3.1 Definitions	1
1.3.2 Acronyms	2
1.4 References	2
2.0 SCM Management	3
2.1 Organization	3
2.2 SCM Responsibilities	3
2.3 Applicable Policies, Directives, and Procedures	4
3.0 SCM Activities	5
3.1 Configuration Identification	5
3.1.1 Software Identification Scheme	5
3.1.2 Documentation Identification Scheme	6
3.1.3 Software Media Identification Scheme	6
3.1.4 Software Enhancement/Problem Report (SEPR) and Software Change Request (SCR) Numbering Scheme	7
3.1.5 Software Libraries	7
3.1.6 Storage and Handling	8
3.1.7 Offsite User Access Control	9
3.2 Configuration Control	9
3.2.1 Placing Software Items Under Configuration Management	12
3.2.2 Processing Software Enhancement/Problem Reports and Software Change Requests	12
3.2.3 Creation of New Versions/Revisions of Software and Associated Documentation	14
3.3 Configuration Status Accounting	18
3.4 Configuration Audits and Reviews	18
3.5 Interface Control	19
3.6 Subcontractor/Vendor Control	19
4.0 SCM Schedules	20
5.0 SCM Resources	20
5.1 Personnel	20
5.2 Resources	20
6.0 SCM Plan Maintenance	22
Exhibits	

1.0 Introduction

This document is the Software Configuration Management Plan (SCMP) for the Federal Emergency Management Information System (FEMIS) project. FEMIS is an automated decision support system that integrates the planning and analysis, operations and response, and recovery phases of emergency management. It has broad application across the full range of emergency planning and operations. The SCMP will describe the configuration management system to be used for the control and maintenance of the software and associated documentation during development, test, and production of the FEMIS project.

1.1 Purpose

This plan provides information on the requirements and procedures necessary for the configuration management activities of FEMIS. It identifies the software configuration management requirements and establishes the methodology for generating configuration identifiers, controlling engineering changes, maintaining status accounting, and performing audits and reviews during the design and development of software configuration items.

1.2 Scope

This plan defines the Software Configuration Management (SCM) activities necessary for maintaining all support software items and associated documentation being developed, procured, tested, sustained and kept in the production environment for FEMIS. It will apply to all phases of the software development life cycle, up to and including the time of delivery to the customer.

1.3 Definitions and Acronyms

1.3.1 Definitions

The terms used in this plan conform to the definitions found in the Institute of Electric and Electronic Engineers (IEEE) Standard Glossary of Software Engineering Terminology, ANSI/IEEE Std 610.12-1990.

1.3.2 Acronyms

The following acronyms are used within this document:

CCB	Configuration Control Board
CI	Configuration Item
CM	Configuration Management
CSA	Configuration Status Accounting
CSCI	Computer Software Configuration Item
FEMIS	Federal Emergency Management Information System
SCA	Software Change Authorization
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
SCR	System/Software Change Request
SEPR	Software Enhancement/Problem Report
STP	Software Testing Plan

1.4 References

- [1] ANSI/IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.
- [2] ANSI/IEEE Std 828-1990, IEEE Standard for Software Configuration Management Plans.
- [3] PNL-MA-68, Records Management and Document Control.

2.0 SCM Management

2.1 Organization

The FEMIS project organization is designed to ensure clear lines of authority and to provide a framework within which administrative and technical control of software activities can be effectively integrated into a quality product.

Primary authority for managing the project is vested in the Project Manager. Overall responsibility for configuration management will reside with the Project Manager and be performed by an appointed representative, referred to within this document as the CM Coordinator.

The project team will consist of task leaders responsible for the development, enhancement, and integration of their respective software applications within the system. Figure 1 illustrates the main groups and their relationship within the project team.

FEMIS Organization Chart
Figure 1

2.2 SCM Responsibilities

The primary responsibility of the CM Coordinator is to manage the change control process in the software and documentation as it develops. This will entail:

- Maintaining the identification mechanism for all the program software and documentation.
- Managing the change control process for all software and documentation.

- Providing status accounting of all software and documentation.
- Performing configuration audits and reviews as required by the program.
- Establishing a database that may be used for generating status reports.

2.3 Applicable Policies, Directives, and Procedures

Any external constraints placed on this plan by other policies, directives, and procedures shall be identified if and when they arise. Client awareness and agreement to these conditions shall be ensured before their incorporation within this plan.

3.0 SCM Activities

3.1 Configuration Identification

Software configuration identification consists of defining each item of software being developed or maintained by the project, naming and labeling these items, and accurately capturing all the various representations of the software items.

The identification scheme provides for identification of the software, associated documentation and all revisions. The identifier will be included in a header within the source code, if applicable. The header will contain specific configuration management information, such as program identification and history information which consists of date, a brief description of the reason for the change, and the corresponding Software Enhancement/Problem Report (SEPR) and Software Change Request (SCR) number.

The CM Coordinator shall:

- Implement and maintain a uniform positive means for identifying software source code (if applicable), software documentation items (including support software and data), and software media; and ensure adequate traceability for unique software configuration item(s).

3.1.1 Software Identification Scheme

The Systems Administrator shall:

- Ensure that all text files (source code, makefiles, scripts, graphics data) residing on the project development platform are identified through the use of a revision control system. Multiple levels of versions and subversions of identification shall be supported. For the FEMIS project a release number and a version number will be automatically tracked to the filename. Configuration identifiers shall have the following content:

- a. file name = name of the file
- b. release = n-digit release number. The initial release of the software will be release 1.
- c. version = n-digit version number. The initial version will be version 0.

Example: fileA.3.2 where;

fileA = name of the file
3 = release number
2 = revision number

- Ensure that the identification scheme for software documentation be placed in a footer along with the date of issue and pagination.

3.1.2 Documentation Identification Scheme

The Systems Administrator shall:

- Ensure all documentation generated will follow a method of configuration identification that allows project staff to record the purpose and version of a document.

The document configuration identification scheme shall address, at a minimum, the following:

- a. Project name
- b. Item type; the following abbreviations will be used:

CMP = Software Configuration Management Plan
PMP = Software Project Management Plan
QAP = Software Quality Assurance Plan
SDD = Software Design Documentation
SRS = Software Requirements Specification
STD = Software Test Document

- c. Version number
- d. Revision number

Example: FEMIS.SRS.4.2 where;

FEMIS = project name
SRS = item type
4 = version number
2 = revision number

3.1.3 Software Media Identification Scheme

The Systems Administrator shall:

- Identify, by labeling, software or software documentation stored on electronic media. The software media shall be clearly labeled so the contents are readily identified. Labels shall contain relevant information indicated as follows:

Company Name: Self explanatory
Project Name: List the project name
Title: A short descriptive name for media contents, include

configuration items, if appropriate
Date: Origination date
Version/Release: Version/Release of media being submitted
Format: Machine and media format of data (if applicable)
Media: Identification of media and indication of whether it is
a master, backup or copy

Sample label:

Company: Battelle, PNL Date: March 10, 1993
Project: FEMIS
Title: This media contains build 3 of the FEMIS program.
Ver/Rel: 2.0
Format: tar xf/dev/rst0
Media: 3.5" Floppy - MASTER

3.1.4 Software Enhancement/Problem Report (SEPR) and Software Change Request (SCR) Numbering Scheme

The CM Coordinator shall:

- Assign, maintain, and track sequence numbers for SEPRs and SCRs.
- Record and track the sequential numbers in the Software Report Log [Exhibit 1] at the time of assignment.

3.1.5 Software Libraries

Software libraries will be established to facilitate the development and maintenance of the software source code.

The CM Coordinator shall establish the following software libraries:

- The *develop* library will contain preliminary source code and configuration files that are required to run the system. This library is used by developers and has no access controls. The *develop* library will use a revision control system to track changes throughout the development effort.
- The *test* library contains only executables and test data and is under the control of the CM Coordinator. The *test* library serves as the system test area; only those files relating to testing will be kept in this library. Developers cannot change anything within the *test* library without an approved SEPR and have read/execute access only. This procedure is will be under revision control. This will ensure continuity of the test environment.
- The *demo* library will contain executables only and is under the control of the CM Coordinator. This library will serve as the demo area and will be available for general user access.

- The *prod* library contains all executables that are ready to be released and their configuration files. This library is under the control of the CM Coordinator and all users have read/execute access only.
- The *doc* library will contain all on-line FEMIS software documentation including any Configuration Management logs (i.e., Software Enhancement/ Problem Reports, Version/Revision Reports). This library is under the control of the CM Coordinator.
- The *scripts* library will contain all the revision control system script files and general purpose utilities. This library is under the control of the CM Coordinator.

3.1.6 Storage and Handling

During development and throughout the life cycle of the project the project software and documentation shall be backed-up, stored, and handled as follows:

The CM Coordinator shall:

- Create a media backup of the master copy within the *develop* library of the latest version of software source code, software executable image, and data files required to run the software, and other software records normally resident on the computer.
- Control access to the *develop* library copy and media backup copy. The **backup copy shall not be used** unless needed for disaster recovery.
- Maintain a backup copy of all associated software documentation items. The backup copy can be the documentation under the *doc* library and the master would be the hardcopy stored within the project file.

The Systems Administrator shall perform software backups and adhere to the following schedules throughout the lifecycle of the project.

- System backups of changed files and remote backups of developer workstations shall be done daily. These tapes shall be retained for up to one week.
- Full system backups shall be done weekly. These tapes shall be retained for up to one month.
- Full system backups shall also be done monthly. These backups shall be retained for three months and then archived. Archived tapes shall be located remotely from the other backup tapes and stored throughout the life of the project.

The Project Manager or Designee shall:

- Ensure that all software configuration items are maintained as research project records, in accordance with PNL-MA-68, Records Management and Document Control.

3.1.7 Offsite User Access Control

The Project Manager or Designee shall:

- Document authorized offsite user access in a letter to those users, with copies to the CM Coordinator and to research project records.

The CM Coordinator shall:

- Compile an Offsite Software User List for each software configuration item to ensure no unauthorized copies or access to the computer software shall occur.
- Complete the Offsite Software User List for each applicable software configuration item type in the following manner:
 - a. Initial access to a software version shall be indicated in the first column under software version/revision/date of release.
 - b. As a user receives a new software version/revision, the version/revision identifier shall be noted on the next numbered column under software version/revision/date of release.
 - c. When user access is terminated, the date of termination shall be noted under the appropriate column of software version/revision/date of release.

3.2 Configuration Control

Configuration control consists of establishing procedures for 1) placing software items under configuration management, 2) processing software enhancement and change requests, and 3) creating new versions/revisions of software and associated documents. The flow of configuration control is described in the following paragraphs.

The SEPR [Exhibit 2] is submitted to the CM Coordinator and forwarded to the Project Manager for initial screening to determine the appropriateness of the change and the budgetary consequences. Once the SEPR passes this initial screening, the CM Coordinator logs the SEPR. The Project Manager then assigns the appropriate developer to analyze the problem and to propose alternate solutions and their impacts. The Project Manager or a designated authority will either approve and prioritize, disapprove, or defer the proposed change. After this, the SEPR is submitted to the CM Coordinator for disposition.

Once the SEPR has been approved the actual changes made to the code and associated documentation are documented by the appropriate developer using a SCR [Exhibit 3]. The developer checks out the controlled source from the *develop* library. The developer can check out software items within the *develop* library but cannot check them back in. The SCR documents the solution, indicating all changes made, and the method used to verify the solution. Finally, when the software is updated by the developer, the CM Coordinator returns the changed items back to the *develop* library and the SCR may be closed. A SEPR may not be closed until the corresponding SCR is closed. Dispositions are all concluded by informing the originator of the outcome and closeout of the change by the CM Coordinator. Figure 2 describes the flow of the configuration control.

Exhibits 1 through 6 are attached to establish the required content for Configuration Management logs/forms used throughout the life cycle of the project.

Figure 2. SEPR/SCR Flow

3.2.1 Placing Software Items Under Configuration Management

The Developer shall:

- Ensure that the unique software identification number is incorporated into the software documentation, if applicable.
- Check that changes related to the release are published (saved) to the *develop* library.

The CM Coordinator shall:

- Ensure that a unique software identification number has been assigned to the software and/or documentation (see Section 3.1), and identify this on the Software Version/Revision Log [Exhibit 6].
- Ensure that the unique software identification number is identified in the software listing, if applicable.
- Review, verify, and approve any software and/or associated documentation before the item(s) are released, via part 3 of the New Version/Revision Report [Exhibit 5].
- Prepare the executables for test and move the executables into the *test* library.
- Document the following disclaimer, obtain documented approval by the Project Manager, and transmit the disclaimer to the client if software or resulting data must be transmitted to the client prior to the completion of all required verifications and validation:

"(Program Name)" is being transmitted for your use prior to completion of final product review actions. No assurance is expressed or implied as to the accuracy, completeness or usefulness of the software until formal review actions are complete."

If the software source code and/or executable image must be released to users prior to the completion of all required reviews and testing, the program output should contain an appropriate disclaimer.

3.2.2 Processing Software Enhancement/Problem Reports and Software Change Requests

This section identifies the process for reporting and processing changes for software and associated documentation using the SEPR/SCR process described in this SCMP. A SEPR may be generated by anyone discovering a problem or desiring a change. Proposed changes to released software items or documentation are initiated into the system by a SEPR.

The Originator of the SEPR shall:

- Complete the top portion of the form.
- Submit the SEPR form to the CM Coordinator.

The CM Coordinator shall:

- Ensure that the SEPR is complete, legible, and reproducible. If there is any discrepancy, the form will be returned to the originator for clarification or correction.
- Assign a number to the SEPR.
- Ensure, if the SEPR is disapproved, that a copy of the completed SEPR is forwarded to the originator.
- Enter the number of the SEPR on the Software Report Log.
- Maintain a copy of the SEPR in the file and forward the original to the Developer for evaluation.
- Retain open status on the SEPR until all actions indicated on the SCR have been accomplished and the SCR has been completed.
- Ensure, if the SEPR is approved, that the SCR is assigned and all changes to the software and associated documentation are implemented into a new version of the software and that all appropriate logs updated.
- Move all changed software items into the *develop* library.
- Close SEPR and SCR forms when all actions are complete by signing the SCR form.
- Retain the original SEPR/SCR for the project file.

The Developer shall:

- Ensure, if a software problem is being reported, that the software is run using input supplied with or described on the SEPR and that the appropriate action is accomplished as follows:
 - a. If the error cannot be duplicated with the input provided or it is found that the supplied input is in error, the Developer shall return the SEPR to the CM Coordinator.

- b. If the error occurs as described on the SEPR or if any other errors occur in the run, the Developer shall complete the analysis portion of the SEPR, sign and date the form, and forward the form, input, and output to the Project Manager or designated authority.
- Appropriately check out the software item from the *develop* library. Implement the change and document the changes made on the SCR.
- Check in the items ready to be returned back to the *develop* library.

NOTE: If a problem reported affects software documentation, the document name, number, and version shall be listed on the SEPR and the problem will remain open until all changes have been made to the master documents. If a software problem impacts previous work, the Developer shall initiate a corrective action request.

The Project Manager or Designee shall:

- Determine disposition of the request upon receipt of the SEPR. Disposition of the request shall be entered on the SEPR by signature and date on the form, and returned to the CM Coordinator for processing.
- Assure retention of completed SEPR/SCR forms as project records.

3.2.3 Creation of New Versions/Revisions of Software and Associated Documentation

New versions/revisions of software are created to install the initial version/revision or a set of SEPR/SCRs that have been approved for incorporation into a new software version/revision. A new version/revision of the software documentation corresponding to a new version/revision of the software are created if any SEPR/SCRs listed on Part 1 of the New Version/Revision Report require documentation changes. Software documentation items may be updated independent of, or concurrently, with the software source code. See Figure 3 for New Version/Revision Report flow.

The Developer shall:

- Complete Part 1 of the New Version/Revision Report to identify the base version to be revised, specify changes to be installed in the designated version, and specify testing or other verification activities to be performed in relation to the changes.

Figure 3. New Version/Revision Report Flow

NOTE: Typographical and minor editorial changes to documentation that clarify, but do not change the meaning of the text, may be reviewed and implemented for incorporation into the new version/revision release without formal processing of a SEPR/SCR. These minor changes will be noted on the New Version/Revision Report.

- Ensure that only approved changes are listed in Part 1 for incorporation.
- Ensure that the required test cases include applicable benchmark cases and inputs that generated the problem as documented in the SCR.
- Develop new test as appropriate, and document the reference on the SCR.
- Obtain a controlled copy of the approved base version/revision from the CM Coordinator.
- Revise the software and associated documentation in accordance with Part 1 of the New Version/Revision Report.
- Initiate reviews of the updated software items in accordance with the STP.
- Notify the CM Coordinator that all reviewed and approved software source code items are ready to be tested (see Section 3.2.1) and include software documentation.

The designated Test Team Leader shall:

- Ensure testing is performed in cooperation with the Software Quality Assurance department.
- Assign the appropriate staff member to perform testing activities.
- Ensure that testing is performed from the *test* library and that the required testing is performed as specified in Part 1 of the New Version/Revision Report and the appropriate Software Testing Plan.
- Ensure that all software and/or associated documentation errors discovered during testing are documented on a new SEPR(s) in accordance with Section 3.2.2.
- Incorporate into updated batch files all regression test(s) sets and include files in the demo library and/or CM transmittal.
- Document and transmit to the CM Coordinator the test results, evaluation of results against acceptance criteria, and application logs, if appropriate.
- Notify the CM Coordinator if, during testing, the new version of the software executable image should not be approved for use (i.e., another version needs to be created to fix problems found during testing).

- Concur with testing activities by completing Part 3 of the New Version/Revision Report.

The CM Coordinator shall:

- Ensure access to a controlled copy of the base version of the software and/or associated documentation under revision in the *develop* library for the developer who is creating the new version.
- Assign a unique Software Configuration (ID) to the new version software and/or associated documentation, and provide the ID to the developer so that it can be entered on Part 1 of the New Version/Revision Report (see Section 3.1.1).
- Upon receipt of the New Version/Revision Report and associated updated software configuration items, the CM Coordinator will do the following:
 - a. Examine the software documentation, if the software source code has not been updated, to ensure that all affected documentation has been updated.
 - b. Prepare and provide a copy of the software executable image, if the software source code has been updated, to the test team in the *test* library.
 - c. Notify the Test Team Leader via cc:mail that the version/revision is ready to be tested.
 - d. Ensure testing is performed in cooperation with the Software Quality Assurance department.
 - e. Indicate on the appropriate new version logs and New Version/Revision Report that the software source code has been updated but testing is halted because of problems found, and the new version/revision will not be released for use.
- Identify the new version/revision of software and/or associated documentation and incorporate changes (by SEPR/SCR log numbers) in the Software Version/Revision Log.
- Create backup copies in accordance with Section 6.1.
- Provide a controlled copy of the new version/revision of the software executable image and/or associated documentation to the *prod* library.
- Retain original software version/revision report in the project file.

The Project Manager or Designee shall:

- Approve or disapprove the New Version/Revision Report after the formal

new version or review is complete by completing Part 3 of the New Version/Revision Report.

- Return the original New Version/Revision Report to the CM Coordinator.

3.3 Configuration Status Accounting

Configuration Status Accounting (CSA) is a means by which enhancements/changes and new versions/revisions of software items may be identified and tracked. The FEMIS project will adhere to the CSA practices outlined within this plan.

An internal configuration status accounting system will be established to track problems associated with the software. The status of proposed changes will then be progressively tracked through approval and implementation. These records provide traceability between versions of software and associated documentation. Whenever possible, this tracking system will be automated.

The CM Coordinator shall:

- Generate and maintain a Software Report Log to show the status of proposed changes, status of approved changes, and the implementation status of approved changes.
- Generate and maintain an Offsite Software User List [Exhibit 4] for identification of all offsite users of the software, change notification, and possible problem reporting input.
- Maintain the original SEPRs/SCRs and New Version/Revision Reports.
- Ensure that the New Version/Revision Report, at a minimum, includes an inventory of the specific versions/revisions of the software configuration items released. All changes shall be identified.
- Generate and maintain a Software Version/Revision Log for each software item including support software, so that each version/revision is uniquely identified and its contents documented.

3.4 Configuration Audits and Reviews

Configuration audits and reviews are used to determine to what extent the actual software and documentation reflect the required physical and functional characteristics. They will determine what software and documentation is present and what should be present.

The CM Coordinator shall ensure that periodic configuration audits and reviews are completed in the manner and times required and defined by the customer

and/or government regulations. These audits and reviews will evaluate programming software, third party software, operational software and data, test software and data, procedures and control systems, and applicable documentation.

At the conclusion of each configuration audit or review, the CM Coordinator shall ensure that a report outlining the observed differences and the resolutions to these differences is provided to the Project Manager.

3.5 Interface Control

Interface control activities involve coordinating changes to the project Configuration Items (CIs) with those external devices and networks that will interface with FEMIS.

FEMIS will require interfacing to several non-system devices, such as meteorological towers, sirens and other sensing devices. In addition, Wide Area Network (WAN) communications such as dial-up, satellites, and other network links will be employed. As these interfaces are identified and further defined, they shall be incorporated within configuration control.

3.6 Subcontractor/Vendor Control

Subcontractor/Vendor control involves plans for utilizing software products developed outside this project. The FEMIS will be comprised of Commercial Off-the-Shelf (COTS) software applications that will utilize hazard models, meteorological data and possibly other public domain software that must be integrated into a single system. The system will be modular in design and will support a variety of different hazards.

This project does not have responsibility for supporting the development or maintenance and change of COTS software. The CM Coordinator will ensure this software and all other subcontractor/vendor generated software is inspected upon receipt and ascertain that all items are available as required by contract.

COTS products are integral components of this system and will be configuration managed in the same manner as the other software configuration items. Configuration audits and reviews will be conducted on this software, as practical, in accordance with the prescribed procedures outlined within this plan. Upgrades to COTS products will be evaluated by the development team as they occur to determine whether implementation of the upgrade is advised. Configuration control of COTS upgrades will be conducted, as practical, in accordance with the procedures outlined in section 3.2 of this plan.

4.0 SCM Schedules

As stated earlier in this plan, SCM activities will span the entire life cycle of this project. Specific SCM milestones will be correlated to project milestones as the project schedule becomes finalized. Figure 4 is the current FEMIS schedule.

5.0 SCM Resources

5.1 Personnel

The FEMIS Project Manager will ensure that an effective SCM program is established, budgeted, implemented, and maintained. The Project Manager or designated authority is responsible for establishing procedures and plans for the implementation of SCM. These procedures and plans must be coordinated between team members and be in concert with the requirements of the program. The Project Manager shall designate, via memorandum to file, all responsible staff.

The CM Coordinator is the individual responsible for implementing this plan. He is responsible for establishing and maintaining development support libraries, process documentation, generate status accounting indexes and listings, support preparation and distribution of documentation and media, and participate in and provide documentation for software reviews and audits, if applicable. The CM Coordinator shall not be the same person who will be generating the software code and/or documentation.

The Test Team Leader is the individual responsible for ensuring that all testing activities are evaluated, documented, and reported in accordance with the requirements of this procedure and any Software Test Plans.

The Developer(s) is responsible for the generation of Software documentation and source code applicable to the project.

5.2 Resources

The SCM function shall utilize tools as required to perform identification, control, and status accounting of software development. These tools allow for management of software revision and release control in a multi-developer, multi-directory, multi-group environment and accommodate easy tracking of third-party vendor source distributions while maintaining local modifications to earlier versions. These tools can be single or multiple vendor products.

[illegible]

Figure 4. Current FEMIS Schedule

6.0 SCM Plan Maintenance

SCM plan maintenance is necessary to identify the activities necessary to ensure continued SCM planning during the life cycle of the project. It is the CM Coordinator's responsibility to monitor compliance with this plan and ensure changes and updates are implemented to the plan as required.

Reviews of this plan will occur periodically throughout the project life cycle. At a minimum, reviews will occur at the start of each project software phase. At that time, proposed changes, if any, will be evaluated by the CM Coordinator and with the approval of the Project Manager implemented within the plan. All plan changes will be communicated to the project team.

<https://doi.org/10.1093/bioinformatics/btad678>

[illegible]

Page completion certified by:

Configuration Management Coordinator	Date

Instructions for Completing Software Report Log

This exhibit establishes the required content for a Software Report Log. The format may be tailored using a different format (hard copy or electronic) as long as the required content is addressed.

1. Enter unique SEPR number assigned by CM Coordinator on SEPR Form and if applicable the corresponding SCR number.
2. Enter date SEPR was received.
3. Enter the originator of the request (from the SEPR Form..
4. Enter disposition of request (i.e., accept, reject, defer..
5. Enter the date of disposition.
6. Enter the date the users (listed on Software User List) were notified of problem. In some cases such notification may not be necessary.
7. Enter the date on which the SEPR was closed.
8. List the Configuration IDs of all affected software configuration items.
9. List the date the change was released in a new version/revision of one or more software configuration items and list the software configuration IDs of the new versions/revisions that contain the implemented change.

Software Enhancement/Problem Report (SEPR)

Enhancement ☐ Problem ☐

Project No. _____

Originator _____ Phone _____ SEPR Number _____

Project Name _____ Origination Date _____

Problem Description _____

Category: ☐ Software ☐ Documentation ☐ Design

Affected Module(s) _____ Version _____

Document Affected ☐ Y ☐ N Document _____ Section No. _____

Analysis

Name _____ Organization _____ Phone _____

Date Assigned _____ Date Complete _____

Possible Resolutions (Include Impacts) _____

Action Required _____

Analyst's Signature _____

Disposition

Project Manager/Designated Authority _____

☐ Approved ☐ Disapproved ☐ Deferred

Date _____

Software Enhancement/Problem Report (SEPR)
Continuation Sheet

SCR Number _____

Instructions for Completing Software Enhancement/Problem Report

This exhibit establishes the required content for a Software Enhancement/Problem Report (SEPR). The format may be tailored using a different format (hard copy or electronic) as long as the required content is addressed.

1. Originator enters the appropriate number of pages to the SEPR.
2. Originator indicates whether the SEPR is a problem or enhancement by checking the appropriate box.
3. Enter the name and phone of the originator of the SEPR.
4. The CM Coordinator enters the next available SEPR number.
5. The originator enters the date the SEPR was generated.
6. The originator describes the problem or enhancement being reported. Provide enough detail of problems so that the problem may be recreated by the assigned developer. If more room is required use the SEPR continuation sheet [see page 2 of this exhibit].
7. The originator indicates the category of the problem by checking the appropriate box. More than one box may be checked.
8. The originator indicates the affected module(s) and the appropriate version number, if known. If the originator does not know this information leave blank and the CM Coordinator will ensure the information is listed.
9. The originator indicates whether or not documentation has been affected. If yes, the document name and section number is indicated.
10. CM Coordinator enters the name, organization, and phone of the assigned analyst.
11. Enter the date the SEPR was assigned to the analyst.
12. The analyst enters the date the analysis was completed.
13. The analyst describes the resolution of the SEPR. If more space is required, use the SEPR continuation sheet [see page 2 of this exhibit].
14. The analyst describes any actions that may need to be completed as part of the SEPR completion.

15. The analyst signs the Analysis Section of the SEPR when completed, usually the code custodian.
16. The Project Manager/Designated Authority signs the SEPR for disposition.
17. The Project Manager/Designated Authority indicates the disposition by checking the appropriate box.
18. The Project Manager/Designated Authority enter the date of the disposition. If not accompanied by an SCR, this is the closure date of the SEPR.

Software Change Request (SCR)

Solution

Ref. SEPR Number _____ SCR Number _____

Software Modules Corrected _____

Document Corrected _____ Version _____ Section No. _____

Corrective Solution _____

Impacts _____

Implemented by _____

Internal Verification

Test Conductor _____ Date Verified _____

SVVP/SVR References _____

Verification Description _____

Authorization to Update Software Configuration

CM Coordinator _____ Date _____

Closure Date _____

Software Change Request (SCR)
Continuation Sheet

SCR Number _____

Handwritten continuation of the Software Change Request (SCR) details, consisting of multiple lines of text.

Instructions for Completing the Software Change Request

This exhibit established the required content for a Software Change Report (SCR). The format may be tailored using a different format (hard copy or electronic) as long as the required content is addressed.

1. The Developer enters the appropriate number of pages to the SCR.
2. The Developer enters the appropriate SEPR number.
3. The CM Coordinator enters the corresponding SCR number.
4. The Developer lists the ID name of all modules that have been corrected due to the SEPR. If additional space is needed, use the SCR continuation sheet [see page 2 of this Exhibit].
5. The Developer lists all documents that have been corrected, including version and section number. If additional space is needed, use the SCR continuation sheet [see page 2 of this Exhibit].
6. The Developer describes the solution(s) being implemented. If additional space is needed, use the SCR continuation sheet [see page 2 of this Exhibit].
7. The Developer describes the impact(s), if any, of the changes being implemented.
8. The Developer signature indicates that the changes have been completed.
9. Enter the name of the person performing the testing and the date the testing was completed.
10. List the test cases used (by reference to the SVVP section and/or SVR number). If this is a new test case, indicate on the form and add to the regression test case file.
11. Describe the verification activities. If additional space is needed, use the SCR continuation sheet [see page 2 of this Exhibit].
12. The CM Coordinator signs and dates the SCR once the software configuration has been updated.
13. The CM Coordinator enters the date the SCR is closed. Usually when all logs have been updated.

Offsite Software User List

This exhibit establishes the required content for a Software User List. The format may be tailored using a different format (hard copy or electronic) as long as the required content is addressed.

Project No.: _____

Project Name: _____

Configuration Item Type: _____

NAME AND ADDRESS OF USER	CONFIGURATION ID		NO./DATE OF RELEASE	
	1	2	3	4

Page Completion Certified By: _____
Configuration Management Coordinator Date

New Version/Revision Report

This exhibit establishes the required content for a New Version/Revision Report. The format may be tailored using a different format (hard copy or electronic) as long as the required content is addressed.

PROJECT NO.: _____

PROJECT NAME: _____

BASE VERSION SOFTWARE CONFIGURATION IDs _____

NEW VERSION SOFTWARE CONFIGURATION IDs _____

(Request ID from Configuration Management Coordinator)

PART 1 - To be completed by the Developer.

Computer Type and Operating System: _____

List SEPR/SCR numbers (if any) to be installed:

List the test case(s) to be used (if any) (can reference STP if appropriate):

Developer

Date

New Version/Revision Report (contd)

PART 2 - To be completed by the Configuration Management Coordinator.

All new versions/revisions of configuration items identified in Part 1 have been received and are ready for controlled release. ☐ YES ☐ NO

Comments:

Configuration Management Coordinator _____ Date _____

PART 3 - To be completed by the project manager/designated authority.

The revised version of the software configuration item is:

☐ Approved for Use ☐ Disapproved for Use -

Release Date: _____

Reviewed and Approved By: _____
Project Manager or Designee Date _____

Concurrence By: _____
Test Team Leader Date _____

Software Version/Revision Log

Page 1 of 1
Exhibit 6

This exhibit establishes the required content for a Software Version/Revision Log.

The format may be tailored using a different format (hard copy or electronic) as long as the required content is addressed.

Project No.: _____

Project Name: _____

Configuration Item Type: _____

Configuration Item ID	Creation Date	Release Date	Description (Include file location if applicable)

Page Completion Certified By: _____

Configuration Management Coordinator

Date

DISTRIBUTION

No. of
Copies

No. of
Copies

OFFSITE

ONSITE

- 2 DOE/Office of Scientific and
Technical Information
- 2 Federal Emergency Management
Agency
Attn: Ed Corvi
500 C Street SW
Washington, DC 20472

- 1 DOE Richland Operation Office
R. B. Goranson
- 13 Pacific Northwest Laboratory
T. F. Coonelly (3)
L. H. Gerhardstein
J. A. Hucks
W. D. Millard
F. E. Strycker
B. M. Wood
C. Winters
Publishing Coordination
Technical Report Files (3)

**DATE
FILMED**

11 / 23 / 93

END