

**Subsurface Transport Over Multiple Phases
(STOMP)
Software Configuration Management Plan
Rev. 1.5**

M.D. White
V.L. Freedman

December 2012

Prepared for
the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest National Laboratory
Richland, Washington 99352

This document maintained electronically at
<https://spcollab.pnnl.gov/sites/stompshare/Shared%20Documents/Forms/AllItems.aspx>

Revision Log

Revision	Date	Authors	Summary of Changes
1.0	October 2006	V. Freedman, M. White	Draft Software Configuration Management Plan
1.1	January 2007	V. Freedman	Minor updates to procedures in response to review comments
1.2	April 2007	V. Freedman	Updated use of revision log
1.3	June 2007	V. Freedman	Inserted reference to external user list documentation
1.4	October 2010	V. Freedman	Updated to reflect current documentation process
1.5	November 2012	V. Freedman	Updated to reflect current documentation process
1.5	December 2012	V. Freedman	Added graphic to show branches of STOMP

Approvals

Approved: 
Mark White
Software Configuration Manager/Code Custodian

12/12/2012
Date

Approved: 
Kary Cook
Software Quality Engineer

12/11/2012
Date

Table of Contents

1.0	<u>INTRODUCTION</u>	1
1.1	PURPOSE	1
1.2	DEFINITIONS	1
1.2.1	STOMP UTILITIES.....	1
1.3	REFERENCES	2
2.0	<u>SOFTWARE CONFIGURATION MANAGEMENT</u>	2
2.1	ORGANIZATION AND RESPONSIBILITIES	3
2.2	ORGANIZATION AND RESPONSIBILITIES	3
2.2.1	SOFTWARE CONFIGURATION MANAGER (SCM)	3
2.2.2	CODE CUSTODIAN (CC)	4
2.2.3	DEVELOPERS FOR STOMP AND RELATED SOFTWARE.....	4
2.2.4	TESTERS	5
2.2.5	TRAINING	5
2.3	ADDITIONAL CONSTRAINTS	5
3.0	<u>SOFTWARE CONFIGURATION MANAGEMENT ACTIVITIES</u>	5
3.1	COMMUNICATIONS	6
3.2	SOFTWARE CONFIGURATION IDENTIFICATION	6
3.2.1	IDENTIFYING CONFIGURATION ITEMS	7
3.2.2	BASELINES CHANGES.....	8
3.2.3	LIFE-CYCLE MODEL.....	8
3.2.4	PROJECT FILE DIRECTORY STRUCTURE.....	8
3.2.5	STORAGE AND HANDLING.....	9
3.3	SOFTWARE CONFIGURATION CONTROL	9
3.3.1	PLACING ITEMS UNDER SOFTWARE CONFIGURATION MANAGEMENT	10
3.3.2	PROCESSING SOFTWARE CHANGE REQUESTS.....	10
3.3.3	DOCUMENTING RELEASES	11
3.4	SOFTWARE CONFIGURATION STATUS ACCOUNTING	11
3.5	SOFTWARE CONFIGURATION MANAGEMENT ASSESSMENTS AND REVIEWS	11
4.0	<u>SOFTWARE CONFIGURATION MANAGEMENT SCHEDULES</u>	11
5.0	<u>SOFTWARE CONFIGURATION MANAGEMENT RESOURCES</u>	12
5.1	PERSONNEL	12
5.2	SOFTWARE AND HARDWARE	12
6.0	<u>SCMP MAINTENANCE</u>	12
7.0	<u>STOMP ROLES AND RESPONSIBILITIES</u>	14

List of Figures

FIGURE 1: STOMP CODE MAINTENANCE SHOWING NQA-1 AND CO2 PRODUCTION CODES AND THE MAIN DEVELOPMENT PATH.....	6
FIGURE2: SOFTWARE LIFE-CYCLE MODEL.....	8

1.0 Introduction

This document describes the software configuration management activities to be performed in support of the Subsurface Over Multiple Phases (STOMP) simulator and its associated utilities. This Software Configuration Management Plan (SCMP) plan applies to all software listed in the Software Test Plan (STP), and is referred to as STOMP and related software in this document.

1.1 Purpose

This SCMP provides information on the requirements and procedures necessary for the software configuration management activities of the STOMP simulator. This SCMP identifies the software, hardware, and documentation requirements for software configuration management and establishes the methodology to generate configuration identifiers, manage engineering, scientific and input/output formatting changes, maintains status accounting, and perform assessments and reviews during requirements analysis, design, development, and maintenance of the Software Configuration Items (SCIs).

1.2 Definitions

Glossary

Term	Definition
CC	Code Custodian
CVS	Concurrent Versions System (configuration management software)
CSA	Configuration Status Accounting
DOE	Department of Energy
OTD	Office of Technology Development
PNNL	Pacific Northwest National Laboratory
RIDS	Record Inventory and Disposal System
RL	Revision Log
SCI	Software Configuration Item
SCM	Software Configuration Manager
SCMP	Software Configuration Management Plan
SCR	Software Change Request
SDD	Software Design Document
SRS	Software Requirements Specification
STOMP	Subsurface Over Multiple Phases
STP	Software Test Plan

1.2.1 STOMP Utilities

In addition to the STOMP simulator, a STOMP preprocessor (sTeP) can be used to set array sizes for STOMP compilation when compiling a static executable. The preprocessor is not needed when compiling STOMP as a dynamic executable. Three perl scripts can be used to extract STOMP output into Tecplot®, Gnuplot and Surfer formatted input files. All these processors are maintained under CVS control in the same manner that the STOMP source code is maintained.

1.3 References

- Bramley, R and X. Wang. 1995. SPLIB: A Library of Iterative Methods for Sparse Linear Systems.
- Balay, S., K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith and H. Zhang. 2004. PETSc User Manual. ANL-95/11 – Revision 2.3.2, Argonne National Laboratory.
- Cederqvist et al., 1993, Version Management with CVS for CVS 1.10.8. Available on the Internet at http://www.wincvs.org/howto/cvscod/cvs_1.htm#SEC1.
- Fayer, M.J. 2012. Software Quality Plan (SQP) for Subsurface Transport Over Multiple Phases (STOMP).
- Nichols, W.E., N.J. Aimo, M. Oostrom and M.D. White. 1997. STOMP Subsurface Transport Over Multiple Phases Application Guide. PNNL-11216, Pacific Northwest National Laboratory, Richland, Washington.
- Ward AL, MD White, EJ Freeman, and ZF Zhang. 2005. STOMP Subsurface Transport Over Multiple Phase Addendum: Sparse Vegetation Evapotranspiration Model for the Water-Air-Energy Operational Mode. PNNL-15465, Pacific Northwest National Laboratory, Richland, Washington.
- PNNL SharePoint Site STOMPshare (<https://spcollab.pnnl.gov/sites/stompshare/>)
- PNNL STOMP website (<http://stomp.pnl.gov>).
- White M.D. and M. Oostrom. 2006. STOMP Subsurface Transport Over Multiple Phases Version 4.0: User's Guide. PNNL-15782, Pacific Northwest National Laboratory, Richland, Washington.
- White, M.D., and M. Oostrom. 2000. STOMP Subsurface Transport Over Multiple Phase: Theory Guide PNNL-11216 (UC-2010), Pacific Northwest National Laboratory, Richland, Washington.
- Zhang, Z.F., V.L. Freedman, G.D. Tartakovsky and M.D. White. 2012. Requirements for Subsurface Transport Over Multiple Phases (STOMP).

2.0 Software Configuration Management

This section describes the software configuration management organization, allocation of responsibility and authority for software configuration management activities, and references the software configuration management policies and directives pertaining to the use of the STOMP simulator and related software (i.e. STOMP utilities).

STOMP and related software used by PNNL for DOE use and its related documentation will remain under software configuration management as long as STOMP and related software is under continued development, in use for supporting site decision making, or further use of the software is contemplated.

2.1 Organization and Responsibilities

The Software Configuration Manager (SCM) and Code Custodian (CC) for the STOMP software is Mark White. PNNL's quality organization has identified Kary Cook to serve as the Software Quality Engineer for STOMP. The following table identifies project software maintenance team members.

Key Staff for STOMP Software Maintenance

Role	Staff	Responsibilities
Software Configuration Manager (SCM)	Mark White	Decision authority to approve software changes
Code Custodian (CC)	Mark White	Software preservation (backup); software configuration management record keeping; source control
Software Developers	Mark White, Diana Bacon, Yilin Fang, Vicky Freedman, Nitin Gawande, Mart Oostrom, Mark Rockhold, Signe White, Mark Williams, Fred Zhang	Software source code development
Software Testers	Mark White, Diana Bacon, Yilin Fang, Vicky Freedman, Nitin Gawande, Mart Oostrom, Mark Rockhold, Signe White, Mark Williams, Fred Zhang	Software testing

2.2 Organization and Responsibilities

Configuration management for the STOMP simulator and related software is designed to ensure clear lines of authority and to provide a framework within which administrative and technical control of development, enhancement and modification activities can be effectively integrated into a high-quality product for scientific research and analysis.

Primary authority for managing the STOMP software is vested in the SCM. The SCM also serves as the CC.

2.2.1 Software Configuration Manager (SCM)

The general responsibilities of the SCM are to control changes in the software and documentation and to ensure testing and verification is performed for each released version and revision. Emphasis is placed on providing guidance on software development, and leading other software developers.

The SCM is responsible for the following:

- maintaining management control of STOMP and related software by
 - approving or rejecting all Software Change Requests (SCRs) based on full consideration of change impacts and software needs

- assuring that all changes made under approved SCRs are properly implemented and independently tested
- controlling the release of SCIs.
- ensuring STOMP and related software developers are trained to the requirements, procedures, and policies of this SCMP before they can serve as change evaluators or implementers
- assisting in software configuration maintenance assessments and reviews, as required

2.2.2 Code Custodian (CC)

The general responsibilities of the Code Custodian (CC) are to maintain and track changes made to STOMP and related software. Emphasis is placed on providing a traceable history of the software and a mechanism for recreating all releases.

The CC is responsible for the following:

- ensuring that the CVS code repository is established and maintained to support software
- assuring that an SCR log has been successfully completed to show the status of proposed, approved, and concluded changes
- configuration maintenance activities under this SCMP
- maintaining regression tests to be run when the baseline changes
- ensuring that backups of software, data, and records are maintained
- establishing the software configuration management system log and records

The STOMP SharePoint will be used by the CC and the STOMP development and testing team for the following:

- ensuring that SCI naming and numbering conventions, as described in Section 3.2, “Software Configuration Identification” are followed
- maintaining the original SCRs and Revision Logs
- maintaining QA training records
- maintaining a STOMP Users List

Procedures for managing change are defined in Section 3.0, “Software Configuration Management Activities.” The SCR form on the SharePoint site will be used to track the status and document the sequence of steps required to complete the configuration change.

2.2.3 Developers for STOMP and Related Software

Developers for STOMP and related software are responsible for ensuring that changes to SCIs managed under this plan are only undertaken in accordance with the policies, procedures, and requirements of this SCMP. Developers must receive training on this SCMP before they are allowed to perform duties as implementers of changes approved in the SCR process. Developers will implement changes for approved SCRs as assigned by the SCM. Developers may run applicable baseline tests, based on the STOMP mode they are running, but independent testing will be performed on their implementation of SCRs.

2.2.4 Testers

The general responsibilities of the Testers are to test STOMP and related software and verify that SCRs are implemented based on new requirements. Testers are responsible for creating and implementing Test Plans prior to testing. Testers are also responsible for generating Test Reports with the results of the testing and uploading these documents to the STOMP SharePoint.

2.2.5 Training

Staff members charged with responsibilities under this plan will receive training to learn the requirements, procedures, and policies of this plan. The SCM is responsible for assigning training to STOMP team members. Completion of training will be documented and maintained on the STOMP SharePoint site.

2.3 Additional Constraints

Identification of all files, documentation, media and software directories follow the procedures outlined in Section 3.1. Any additional external constraints placed on this SCMP will be identified as they arise. These include additional client requirements, additional standards, or other constraints. Client awareness of and agreement to such constraints and their impact on the project will be obtained before they are integrated into this SCMP.

3.0 Software Configuration Management Activities

STOMP software is widely adopted and applied both in and out of the Laboratory, at Hanford and at other sites worldwide. Therefore the quality of the software is expected to be above reproach. Every effort will be undertaken to minimize adverse outcomes.

The primary means to minimize the risk of a software error of consequence are:

- Strict adherence to this SCMP,
- Strict adherence to the Software Test Plan, and
- Timely identification, response, and communication regarding software errors and anomalies discovered by PNNL staff involved in use, maintenance, and development of the STOMP software.

This section identifies all functions and activities required to manage the software configuration of STOMP and related software products and documentation.

Software configuration management activities are grouped into four general areas:

- software configuration identification
- software configuration control
- software configuration status accounting
- software configuration management assessments and reviews.

3.1 Communications

The SCM is responsible for timely communication of training requirements, risks/hazard exposure, and assignment of commitments/performance of software development and testing staff.

If adverse impacts to completed, reported software calculation results are discovered, the SCM will investigate the impact of the error, and if found consequential, report the error and impact to the concerned project manager(s) and product line manager(s). STOMP users will be notified via the STOMP RSS feed set up on SharePoint. Subscription to this feed is required to receive STOMP announcements.

3.2 Software Configuration Identification

STOMP source code is dynamic research oriented code, as new capabilities are continuously being added to the simulator. Because STOMP source code evolves due to research needs, a separate version of the code will be created that will remain unchanged for applications that require rigorous testing. This static version of the code will be thoroughly tested according to the Software Test Plan, and only bug fixes will be applied to this version of the code on an as needed basis. For the -W, -WAE, -WO and -WOA modes, this will be referred to as the NQA-1 version of the source code. As time and resources permit, the research version of STOMP will be tested to establish a new static, NQA-1 version of the code that can be used for applications that require this level of software compliance. For the CO2 modes, -CO2 and -CO2e, a separate version of the code will also be created and be referred to as the CO2 version of the source code.

Similar to the NQA-1 production code, the research version of STOMP will be tested to establish a new static CO2 code in the future, time and resources permitting. Figure 1 shows how these production codes and main development path will be maintained.

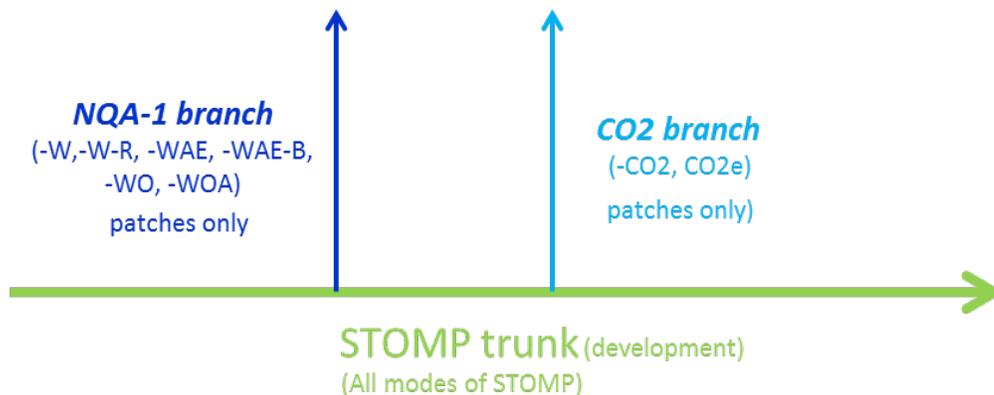


Figure 1: STOMP Code Maintenance Showing NQA-1 and CO2 Production Codes and the Main Development Path

Each independent source file, which may contain one or more subroutines for a particular operational mode, will be identified as a unique SCI using the scheme described in the following sections. This identification scheme will enable all project team members, including test engineers, quality advocates, and the SCM to locate each SCI quickly and easily.

3.2.1 Identifying Configuration Items

Version control and release management for all software source code will be supported using Concurrent Versions System (CVS) (Cederqvist et al., 1993) as the configuration management system. Modified source code will be checked into CVS by the CC as part of the SCR process.

A system of revision numbers will be implemented by CVS for all files, incrementing each revision number by one-tenth with each change approval by the SCM. For example, the first source file checked into CVS is assigned a revision number of 1.1. If a subroutine within the source file is revised, then the newly assigned revision number is 1.2, and is written as a header in the source file. Because revision numbers are tracked by source file, this also changes the revision number for subroutines within the same file that have been unaffected by the change. The revision number for each STOMP file accessed during execution will be printed to both the output file and screen at the end of the simulation. The new revision number will also be recorded in the SCR form during change implementation.

Because the user is responsible for creating an executable, source code is released directly to users. Complete specification of a version of STOMP requires listing the revision numbers of each source file in addition to the date the source was released. The source release date serves as a STOMP version number, as revision numbers of individual source files may change between release dates.

STOMP source code contains a direct solver that can be used on small problems with low memory requirements. STOMP may also be executed with the SPLIB solver (Bramley and Wang 1995) and the PETSc (Balay et al. 2006). The SPLIB solver can be obtain via ftp at <ftp://ftp.cs.indiana.edu/pub/bramley/splib.tar.gz>, and PETSc is available at <http://www.mcs.anl.gov/petsc>.

3.2.1.1 Identifying Software Documentation

Documents will be divided into three categories: 1) principal documents, 2) addendums and errata and 3) supporting documentation. Principal documents will be identified by version numbers only. Currently, principal documents are the STOMP User's Guide (White and Oostrom 2006), the STOMP Theory Guide (White and Oostrom 2000) and the STOMP Application Guide (Nichols et al. 1997). Addendums and errata, published or unpublished, will typically be used to document major revisions, developments or modifications. These documents will be identified with revision and version numbers. Currently, the theory guide is updated with addendums, whereas the user's guide is updated as needed as a PNNL report.

Supporting documentation for STOMP includes the Software Quality Plan, Requirements and Design Documents, this SCMP, and the Software Test Plan. This documentation will be maintained in Microsoft Word with version control (provided via the SharePoint site) and maintained by the CC. As new documentation is developed, all older versions of documents will be maintained electronically, as archival material. To create a new version, the latest version will be copied; all changes accepted using the "Track Changes" features of Word, and the version number incremented in the title page and the file name. Change tracking will be used to record changes since the last version. The older version will be moved into an archive subdirectory.

3.2.1.2 Software Source and Documentation Storage and Preservation

Storage and preservation project software files will be the responsibility of the CC. All source code will be maintained electronically on kahuku.pnl.gov. All documentation pertaining to STOMP and related

software configuration management, user guides, software requirements, and related documents will also be maintained electronically on the STOMP SharePoint site.

The contents of the kahuku will be backed up by the following means:

- The directories including the CVS repository are backed up daily using PNNL’s backup service WBR Mac. These backups are incremental, only backing up changes that have occurred since the last full backup.

3.2.2 Baselines Changes

Each software module will be subject to configuration baselines changes as they are affected by each completed software change request implementation. These baseline changes will be documented for each final code release.

3.2.3 Life-Cycle Model

The life-cycle model is established to ensure the proper evolution of SCIs from concept to release of the software. The model shown in Figure 2 defines how each SCI progresses through the software management and development process.

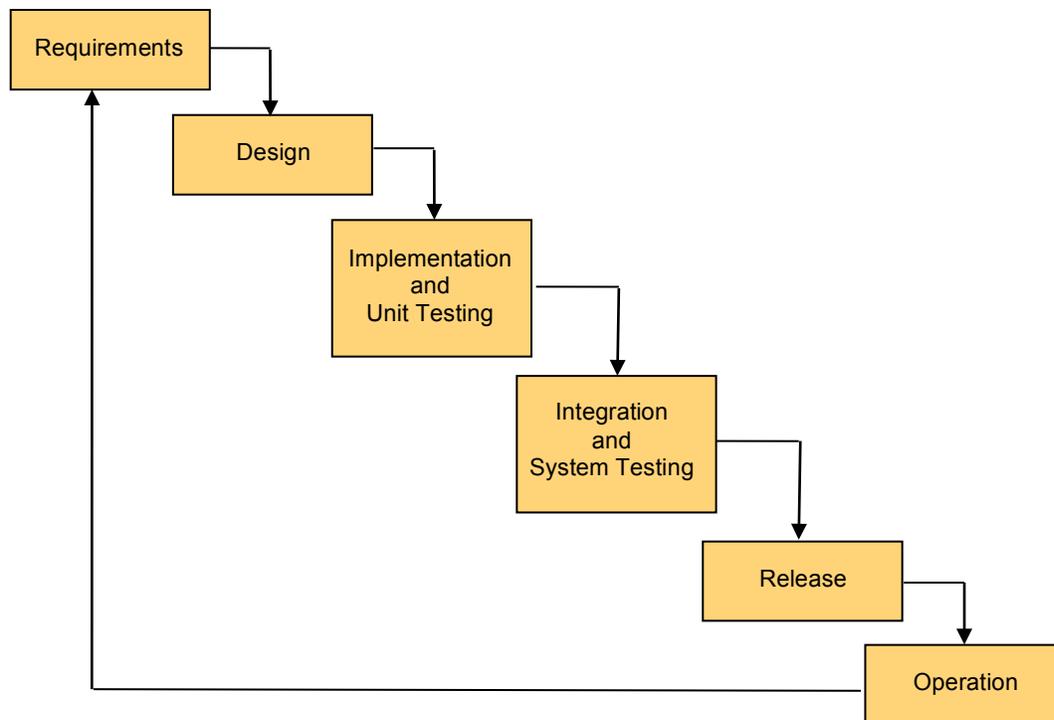


Figure2: Software Life-Cycle Model

3.2.4 Project File Directory Structure

A standard hierarchical directory structure will be maintained for STOMP and related software to facilitate the development and maintenance of electronic documentation and software source code.

All STOMP source code is stored on *kahuku*, which is only accessible to the CC. The directory structure

for the NQA-1 branch, CO2 branch, STOMP trunk, and development code are shown in the following layout:

```
/Users/d3c002/stomp/code/src_nqa1_091911_patches  
/Users/d3c002/stomp/code/src_co2_patches  
/Users/d3c002/stomp/code/src  
/Users/d3c002/stomp/code/src_dev
```

STOMP utilities used for pre- and post-processing data are stored main directory for each version of the STOMP source code.

Software documentation (e.g., manuals, bug fixes, etc.) is accessible through the STOMP web site at and on the STOMP SharePoint site.

STOMP test cases are also described on the STOMP SharePoint Site. STOMP modes that have met the NQA-1 testing requirement are listed on the site. All input and output files are also accessible at the web site so that users can verify their source code with the test cases. A user may also identify which capabilities have been covered by internal testing.

Other directories may be permitted but the directories identified in this plan are considered core directories that must be maintained.

3.2.5 Storage and Handling

All project SCIs will be identified to ensure that the items are properly stored for traceability, defensibility, and reproducibility. The Code Custodian will ensure that project SCIs are stored and handled appropriately. This includes storage of documents and electronic media, marking and labeling of SCIs, and retention periods. The released software and CVS archive, which can be used to recreate unique versions/revisions of the software will be maintained on the kahuku workstation. Backups of these files occur as described in Section 3.2.1.2.

3.3 Software Configuration Control

Software configuration management consists of establishing procedures for

- placing configuration items under software configuration management
- processing Software Change Requests (SCRs)
- creating new versions of software and documentation.

3.3.1 Placing Items under Software Configuration Management

Documents are placed under configuration control upon publication release or final review of unpublished documents (e.g., letter reports, addendums, errata). Software documentation will be subject to manual version control, as described in Section 3.2.1.1. Documents are stored in electronic form on the SharePoint site, and paper copies are stored in project files.

Source code modules are placed under the SCM when they become stable and have passed testing. All files related to a source code file or module, such as header and include files, are promoted along with the corresponding source code.

3.3.2 Processing Software Change Requests

Requests for software changes for the production versions of the code (e.g., NQA-1 and CO2) are formally documented via a SCR form that can be obtained on the SharePoint site. Code revisions are also tagged with a description of changes when source code is checked into CVS. Final testing and verification of the changes are performed for all software changes.

If the SCR is a request for a bug fix, the SCM is responsible for the SCR approval. If the SCR is a development task, the SCM may evaluate the request independently, or may opt to alert members of the STOMP development team that an SCR needs to be evaluated for approval. This decision process is at the SCM's discretion. The SCM or the members of the STOMP development team will assess the following:

- Change Evaluation: is the proposed change reasonable and practical?
- Software Design Requirements: is the proposed change consistent with the Software Requirements Document, as outlined in Requirements Document?
- Criticality: is the proposed change critical to completing client deliverables?
- Time Required: how much time will the proposed change require for implementation?
- Configuration Control: will this change create a necessity to constitute a new release?
- Quality Control: How will this change impact the existing code and how much regression testing will be required?

If the change is approved, the SCM will assign it to one or more software development team members for implementation. The software developer(s) will make the software changes proposed in the SCR and perform unit testing as needed on the changes, and notify the CC that the code is ready to be checked in. The CC will then check the updated source code into the CVS repository.

Tests identified in the STOMP software test plan are performed once a new release is updated. The amount of testing required will be evaluated by the STOMP developer that implemented the change, which may require that testing be performed on all operational modes in the test suite. A STOMP Tester will verify the new source by running the new source through the appropriate test cases. For a new functionality, new test cases will be developed and verified by independent testing. When verification is complete, the CC will check in the revised source code to CVS and will release the source to the STOMP user.

When new source is released, the CC may complete the SCR process by notifying users of the final status of the SCR, which may include:

- SCR originator (person who requested one of the changes implemented in the new release)
- All STOMP development team members

- All STOMP users

No forms are required to request a software change request for the research branch of the code. For the research branch, software changes are recorded in the CVS Revision system by tagging the revised code with text that describes the code changes.

3.3.3 Documenting Releases

The information needed to reconstruct any release (that is, the state of STOMP tools on a given date) will be available through use of the CVS configuration management software. The CVS tools enable the CC to extract source code from the repository by date, which can then be used to build any STOMP executable in effect on the specified date. The full specification of a release of will include the CVS revision number for each source file and the date of the entire source release.

The SCR form will also document release changes, as applicable.

3.4 Software Configuration Status Accounting

Configuration Status Accounting (CSA) is a means by which enhancements/changes and new versions/revisions of configuration items are identified and tracked. A CSA system is already established with the CVS revisioning system. These records provide traceability between versions of SCIs and associated documentation.

3.5 Software Configuration Management Assessments and Reviews

Software configuration management assessments and reviews will be conducted as needed to determine to what extent the actual software and documentation reflect the required physical and functional characteristics. These activities are formal examinations of the *as-built* versus the *as-required* software and documentation. The SCM will participate in all assessments and reviews and record the resulting action items. As each baseline is defined, a quality engineer may also assist with in-process assessments of software configuration management activities.

4.0 Software Configuration Management Schedules

Software configuration management activities will span the entire life of the software. Software development will consist of the following phases:

- Software Requirements Specification
- Design the Capability
- Assemble the Capability
- Testing and Verification
- Reporting
- Closeout*

* The closeout phase of the software will include archiving the source code files and documentation. During retirement, support for STOMP and related software is terminated. STOMP users and developers will be notified electronically (e.g., email, website) to terminate all STOMP and related software usage.

5.0 Software Configuration Management Resources

The SCM will ensure that an effective software configuration management program is established, supported, implemented, and maintained. The SCM or a designated authority is responsible for establishing procedures, plans, and training for the implementation of configuration management.

5.1 Personnel

The SCM is responsible for implementing this plan. The SCM will establish and maintain the development support directories, and process SCM documentation, generating status accounting reports, and preparing and distributing source code. The SCM is also responsible for ensuring that all testing activities are evaluated, documented, and reported according to the requirement of the SCMP. In addition, the SCM will participate in and provide documentation for system development assessments and reviews, if requested.

The development team members are responsible for the generation of software, electronic documentation, and other SCIs applicable to the project under the direction of the SCM.

5.2 Software and Hardware

The SCM and development team members will utilize CVS to perform identification, control, and status accounting to system SCIs.

- Primary STOMP development is carried out on a Mac OS X, 10.4.8 platform.
- STOMP testing is carried out on Linux-based platforms using the Intel FORTRAN compiler (version 10.0), on a RedHat Workstation OS (Linux 2.6.18-274.3.1.el5 #1 SMP Fri Aug 26 18:49:02 EDT 2011 x86_64 x86_64 x86_64 GNU/Linux). If a new compiler is implemented, and/or a new operating system is introduced, then the complete set of tests will be run to establish that STOMP functionality has not been adversely impacted, and establish a new baseline if needed.

Project records are stored and maintained as specified in the project Records Inventory and Disposition System (RIDS).

6.0 SCMP Maintenance

SCMP maintenance is necessary to document configuration management activities throughout the software's life cycle. If any procedures defined in this document are changed, those changes will be reflected in the SCMP, as needed.

It is the SCM's responsibility to ensure the compliance and cooperation of development team members in abiding by this plan. The Software Configuration Manager's responsibility is to monitor compliance, and ensure that changes and updates are reflected in the SCMP, as required.

Reviews of this SCMP will occur periodically throughout the STOMP simulator's life. At a minimum, reviews will occur at the start of each major development phase. At that time, proposed changes, if any, will be evaluated by the SCM and may be approved by the SCM in conjunction with the CCB for implementation. All changes to this plan will be communicated to the development team in a timely manner.

7.0 STOMP Roles and Responsibilities

A list of the STOMP Team and their responsibilities (e.g., developers, testers) is maintained on the STOMP SharePoint site.