IMPORTANT NOTICE: The current official version of this document is available via the Sandia National Laboratories WIPP Online Documents website. A printed copy of this document may not be the version currently in effect.

NUCLEAR WASTE MANAGEMENT PROCEDURE

NP 19-1
SOFTWARE REQUIREMENTS
Revision 17

Effective Date: April 22, 2015

Author:	Jennifer Long	*Original signed by Jennifer Long*	*4/21/15*
	(printed name)	(signature)	(date)

1.0 Purpose and Scope

This procedure prescribes the processes used to qualify and control software used in the Sandia National Laboratories (SNL) Waste Isolation Pilot Plant (WIPP) program. The application of requirements is determined by the intended use of the output from the software. The most rigorous requirements (life-cycle management) are applied to software that is used to demonstrate compliance with disposal regulations [per 40 CFR, Section 194.22 (a) (2) (iv)] or whose output is relied upon to make design, analytical, operational, or compliance-based decisions with respect to the performance of the waste confinement processes (per the CBFO QAPD, Section 6.1.) This type of software is referred to as **Compliance Decision (CD)** software. Examples of this type of software are:

- scientific or engineering software used to assess the performance of a site,
- scientific or engineering software used to analyze data for, or produce input (parameters) to, a performance assessment calculation,
- software that is used in managing information or augmenting mission essential decisions, and
- software used to collect data (e.g., far-field, near-field, engineered barriers), see below.

The qualification process for CD software is described in the body of this procedure.

Because of its impact on data quality and the potential inability to re-collect data, Data Acquisition System (DAS) software must be qualified. If the DAS software is an integral part of an off-the-shelf system and not modified, refer to NP 20-1 (Test Plans under data quality control) for its qualification requirements. If the DAS software is developed or modified for use in the SNL WIPP program, it is considered CD software, and the qualification process described in the body of this procedure must be followed (See Table 1).

Some software that is required to make programmatic decisions such as scoping or screening analyses to develop, implement, or test potential improvements to existing methodology may, with prior approval, need to be used prior to full qualification. This type of software is referred to as **Programmatic Decision (PD)** software. The process for PD software is described in Section 2.4 of this procedure.

Software governed by this procedure shall comply with the applicable requirements of this procedure prior to use.

© 2015 Sandia Corporation

Exempt from this procedure are:

- Commercial-off-the-shelf (COTS) System software such as operating systems, administrative and management systems (database management systems), system utilities, assemblers, compilers, interpreters, etc.
- COTS application software such as Microsoft Office, graphics applications, application utilities, computer-aided software engineering (CASE) tools, etc.
- Software written to conduct simple calculations and other limited applications which can be verified by hand calculations. Use and qualification of these programs is discussed in the analysis procedure NP 9-1.
- Object libraries that are verified/tested as part of qualified software that use them. Object libraries do not require separate validation because their subroutines are validated when the code into which they are linked is validated.

1.1 Definitions

Access Control - The methods established to permit authorized and prevent unauthorized access to software. Controls may consist of restricting access to a computer during off-hours, or providing password security for the computer or the software. These controls may be provided on either a software-specific or a system-specific basis.

Access Control Memorandum - Memorandum which documents access control methods for one or more codes.

Acquired Software - Software brought into the SNL WIPP program, which was not created following the life cycle methodology defined in the DOE/CBFO Quality Assurance Program Document (QAPD). This type of software may have missing life cycle components, and therefore, it needs to be evaluated and qualified prior to use.

Approved Users Memorandum - Memorandum which lists approved users for a particular code.

Code - A computer software item ("code" is used interchangeably with "software").

Code Team/Sponsor - Individual(s) who oversees the Software Quality Assurance (SQA) process for a particular software item.

Code Developer - This individual develops or modifies specific codes at the direction of a subject matter expert.

Commercial Off-the-Shelf-Software (COTS) - Software procured from the commercial sector (e.g., EXCEL, LOTUS, etc.). A characteristic of off-the-shelf software is that it is available for general public use.

Compliance Decision (CD) Software - Software that is used to demonstrate compliance with disposal regulations or whose output is relied upon to make design, analytical, operational, or compliance-based decisions with respect to the performance of the waste confinement processes.

Consistency - Individual requirements are not in conflict with each other.

Data Acquisition System (DAS) Software - Software used to control test equipment, obtain electrical readings from the equipment, convert the readings to scientific or engineering units.

Design Constraints - Describe any functional requirements that will later restrict design options. Examples of this may include operating system, database management system, language, etc. This is often an optional functional requirement category.

Design Document (DD) - A document that describes the major features of the software design: theoretical basis, embodied mathematical model, control flow, control logic, data structure(s), functionalities and interfaces of objects, components, functions, and subroutines used in the software, and the allowed or prescribed ranges for data inputs and outputs in a manner that can be implemented.

Developed Software - Software developed or modified by SNL following life cycle methodology defined in the DOE/CBFO QAPD, as opposed to acquired software.

Functionality - Functional requirements define what the software product must accomplish. They should describe, as applicable: how inputs are transformed into outputs, what inputs are necessary, what outputs are produced, what equations or mathematical techniques are to be implemented by the software, what ranges of inputs can be handled by the software.

Implementation Document (ID) - A document that contains the source code listing (the source code can be stored in a configuration management tool) and documentation of the process used to convert the source code to an executable.

Installation and Checkout (I&C) - The phase of software development where the validated executable code is installed on the production computer and regression testing is conducted to ensure the software performs in the same manner as documented in the Validation Document.

Life-Cycle - A model for software development that starts when a software product is conceived and ends when the software is retired. This model consists of and ensures documentation of technical adequacy.

Major Change: Change with an appreciable effect on the operational characteristics and reliability of the product and its fitness for the intended purpose, relative to the previous version. Major changes typically warrant full re-qualification at least in the areas of code affected.

Manual Inspection - Manual activities which do not involve numerical manipulations. These include visual inspection of output values, table reformatting or plotting, and concurrence with qualitative acceptance criteria such as trends in results due to input parameter variations.

Minor Change: Change with no appreciable negative effect on the operational characteristics and reliability of the product and its fitness for the intended purpose relative to the previous version.

Modeling Software - Qualified software which models physical phenomena, usually by implementation of a system of complex equations.

Object Library – A collection of object files that can be linked to other object files to produce codes. Generally if a subroutine is used by more than one code, its object is stored in a library to avoid redundancy. Object libraries do not require separate validation because their subroutines are validated when the code into which they are linked is validated.

Performance – For software, performance refers to time-related software operations issues, e.g., speed, recovery time, response time.

Primitive Baseline - Software and existing documentation placed under configuration control prior to approval for use.

Production Baseline/Production Software - Baseline software that has been installed and checked out in accordance with this procedure, and therefore approved for use.

Qualified User - A person named in a Qualified User Memo for a specific production baseline Code.

Regression Testing - Software testing conducted to verify that the software produces the same results for a given set of inputs as previously documented.

Relational Database – Comprised of a database engine (e.g., Access, MySQL, etc.) and one or more sets of data, and possibly including management tools such as queries, macros, forms, etc.

Requirements Document (RD) - A software document that contains the requirements that the product must satisfy, including functionality, design constraints, attributes (including acceptance criteria), and external features.

Software Baseline - An item or product that has been formally reviewed and agreed upon, that serves as the basis for further development, and that can be changed only through formal change control procedures.

Software Change Control - The process of proposing, approving, performing, testing, and documenting modifications to production software, system software, and hardware.

Software Configuration Management (SCM) - A system that tracks the software by unique identification, enables the release and retrieval of the software, tracks status and changes to the software and its associated documentation, and defines the code retirement process.

Software Configuration Management Coordinator (SCM Coordinator) - Person responsible for overseeing the operation of the SCM system described in this procedure.

Software Problem Report (SPR) Process - The process of identifying, reporting, and evaluating errors in software. The SPR process ensures that problems with software are identified and documented, all affected parties are notified, and all affected work is identified, evaluated, and revised as necessary.

Software Problem Closure Report (SPCR) Process - The process of closing an SPR. The SPCR process ensures that problems with software are reported as closed when the SPR has reached resolution.

Software QA Plan (SQAP) - A plan for the development of software products necessary to provide adequate confidence that the software conforms to established requirements.

Software Verification and Validation - Verification is the process of determining whether or not the product of a given phase of the software development cycle fulfills the requirements imposed by the current and or previous phase. Validation is the test execution and evaluation process for determining whether the requirements for a software system or component are complete and correct, and the final system or component complies with specified requirements.

System Administrator (SA) - Individual responsible for setting up and maintaining computer hardware, system software, and application software.

System Software - Software that is used exclusively in the preparation, installation, or operation of executable software applications. Examples of such software include operating systems, administrative and management systems, system utilities, compilers, assemblers, translators, interpreters, automated protocols, utilities and tools, and teleprocessing managers.

Technical Review - A documented, critical evaluation of documents, activities, materials, or data conducted to determine the applicability, correctness, adequacy, and completeness of the information submitted for review. Technical reviews must be performed by one or more qualified personnel who are independent of the work being reviewed.

User's Manual (UM) - A document intended for use by a user of the software. The User's Manual contains, as applicable, the software name and version identifier, the platform(s), a statement of functional limitations, instructions that describe the user's interaction with the software, the identification and description of input and output specifications and formats, the valid ranges of input data, descriptions of user messages initiated as a result of improper input and how the user can respond, a description of any required training necessary to use the software, and an explanation of the mathematical model(s).

Validation Document (VD) - A software document that contains the results of the performance verification and validation tests defined in the Verification and Validation Plan (VVP) and evaluation of the outputs of those tests to demonstrate that the software produces valid results for problems encompassing the range of permitted usage as defined by the User's Manual.

Verification and Validation Plan (VVP) - A software document that delineates the test processes and associated acceptance criteria to be performed during validation phase.

Acronyms and definitions for terms used in this procedure may be found in the Glossary located at the Sandia National Laboratories (SNL) WIPP Online Documents web site.

2.0 Implementation Actions

This section contains step-by-step processes for the acquisition, development, maintenance, configuration management, and software problem reporting of CD software. General requirements that apply to the sub-sections are listed below. The user of this procedure should read and understand these steps prior to implementation of any of the sub-sections.

2.1 General Requirements For This Section

1. Quality requirements are summarized in Table 1.
2. The review processes may cause portions of the current phase and/ or previous phases to be modified. In such cases, changes to baseline documents shall be made and verified at the same level of detail as the original document(s).
3. All QA records produced by this procedure are assigned a version identifier composed of two parts as needed, each separated by a period. This system is described below:

   Version **X.Y, e.g., 1.01, 1.00 .** X is the major field. Y is the minor field. Y must have two numerical digits. These version identifiers are changed when new releases of software and/or baseline documents are released. Baseline documents (e.g. RD, VVP, DD, ID, UM, and VD) with the same major version identifiers shall be consistent with each other, however, the major version identifier of the code need not be the same as the major version identifier of the baseline documents.

2.2 Software Qualification

There are two classifications of software which follow life-cycle methodology phases, Acquired and Developed. Table 1 lists applicable requirements for each of these two types of software. Figure 1 shows the documentation flow.

Table 1 Compliance Decision (CD) Software Requirements

PHASE	Planning	Requirements		Design	Implementation		Validation	I&C			Maintenance			Retirement
LOCATION	2.3.1	2.3.2		2.3.3	2.3.4		2.3.5	2.3.6			2.3.7[5]			2.3.8
APPLICABLE DOCUMENT	SQAP[6]	RD[3]	VVP	DD[4]	ID	UM[2]	VD	I[a] & C	AC	AU	CC[a]	SPR[a]	SPCR[a]	
FORMS[1] **NP 19-1-X**	1	2	3	4	5	6	7	8	-	-	9	10	11	-
Acquired	X	X	X	-	[b]	X	X	X	X	X	X	X	X	X
Developed	X	X	X	X	X	X	X	X	X	X	X	X	X	X

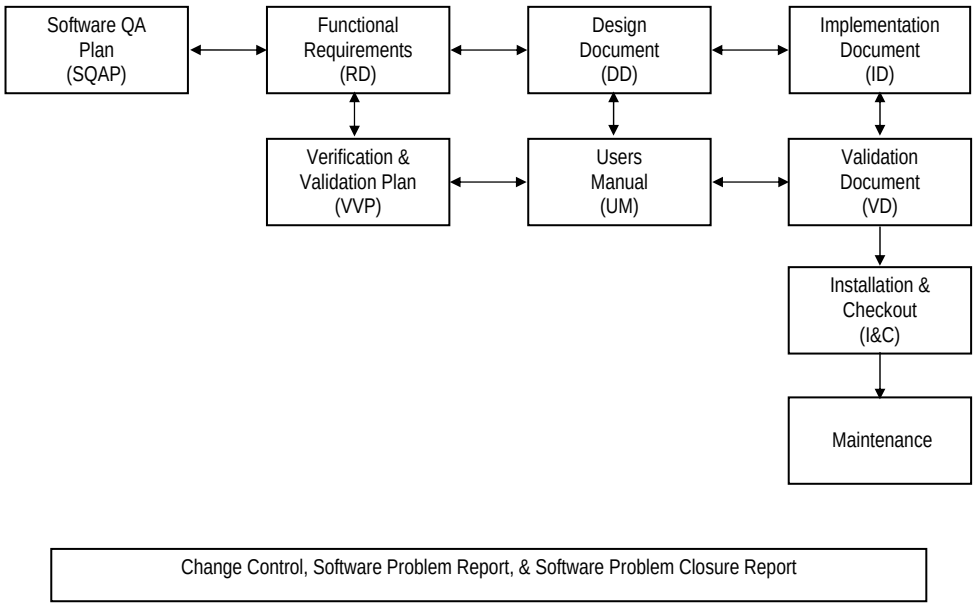
KEY: - indicates that the item is NOT required.
X indicates that the item IS required.
a- the CC , I&C SPR and SPCR are forms only, not documents
AC refers to Access Control Memorandum
AU refers to Approved Users Memorandum
b- Not applicable when the source code is not acquired

TABLE NOTES

1. All form numbers on the table are preceded by NP 19-1-.
2. User Manual (UM) requirements may be fulfilled by referencing and using supplied user instruction publications as long as the supplied documentation complies with the requirements of this procedure.
3. If the requirements of a particular baseline document are provided in multiple documents, a clear path to the fulfillment of the requirements needs to be provided.
4. A Design Document (DD) is not initially required for acquired software. If Acquired software is to be modified and the change is major, a High Level “as built” Design may be developed for the entire existing system depending on the licensing and contract agreements. If the modification is not major in nature, then a detailed design document is required only for the new portions of the design.
5. Change Control (CC), Software Problem Report (SPR), and Software Problem Closure Report (SPCR) are required to document changes to baseline documents, when bugs are discovered, or document the closure of a SPR).
6. Relational Databases may require deviations from the software document requirements due to their lack of typical software attributes, e.g., executable, source code, static size, etc. All deviations must be documented in the SQAP.

Figure 1

Documentation Development Flow



2.3 Software Life Cycle Phases

The life-cycle phases described in this procedure are:

- Planning,
- Requirements,
- Design,
- Implementation,
- Validation,
- Installation and Checkout,
- Maintenance, and
- Retirement.

The activities associated with the evolution of the software shall use an iterative or sequential approach.

Note: Each phase follows the process flowchart in Appendix A.

2.3.1 Planning Phase

Software QA Plan (SQAP) is produced during this phase for new software development. Software being revised that is currently under configuration control and developed within the scope of these QA requirements will not require another standalone SQAP unless the current SQAP omits required elements.

The SQAP shall identify:

- The software to which it applies, objectives of the software, problem statements, and necessity of the development action. The documents to be prepared, reviewed and maintained during the software life cycle, and their relationship to QA measures defined in this procedure. For acquired software, a comparison of the adequacy of the existing documentation with the software life cycle requirements of this procedure will be included in the SQAP.
- If any deviations from the documentation required by NP 19-1 are anticipated, (e.g., a database may not require an Implementation, Verification and Validation Plan, User's Manual and Validation Document), the SQAP should contain a detailed explanation of how the intent of lifecycle reporting will be met. For efficiency, documents may be merged into combined reports.
- The organizations and/or individuals responsible for performing the work and achieving software quality and their tasks with a schedule for qualification and responsibilities.
- The standards, conventions, techniques, or methodologies that guide software development, as well as the methods used to assure implementation of requirements
- The procedure(s) (NPs, SPs, etc.) used for establishing and maintaining the integrity of data, embodied mathematical models, and output files
- The process for reporting and documenting software discrepancies, evaluating the impact of errors on previous calculations, and determining the appropriate corrective action.

Following the development of the SQAP, no strict sequence of performing activities is required provided that all specified requirements for each phase are met and the intent of the requirements are not subverted.

The SQAP may be written for an individual code or a set of codes. It shall be developed by the Code Team/Sponsor and shall be approved (by signature) by the Responsible Manager, Technical Reviewer(s), QA Reviewer, and the SCM Coordinator following the process described in Appendix A, using the phase criteria listed on the Software QA Plan Criteria Form NP 19-1-1, (Appendix B).

2.3.2 Requirements Phase

The following documents are produced during this phase:

- Requirements Document (RD) – defines the requirements that the proposed software must satisfy, and
- Verification and Validation Plan (VVP) - identifies tests to be performed and associated acceptance criteria to ensure verification of each software development phase and validation of the entire software baseline.

The Code Team/Sponsor shall develop the RD and VVP using the phase criteria listed on the Requirements Document Criteria Form NP 19-1-2, (Appendix C) and Verification and Validation Plan Criteria Form NP 19-1-3, (Appendix D). The RD and VVP shall be approved (by signature) by the Responsible Manager, Technical Reviewer(s), QA Reviewer, and the SCM Coordinator following the process described in Appendix A.

2.3.3 Design Phase

The Design Document (DD), produced during this phase, provides the following information (as applicable):

- Theoretical basis (physical process represented),
- Mathematical model (numerical model),
- Control flow and logic,

- Data structures,
- Functionalities and interfaces of objects, components, functions, and subroutines,
- Ranges for data inputs and outputs, in a manner that can be implemented into software.

The Code Team/Sponsor shall develop the DD using the phase criteria listed on the Design Document Criteria Form NP 19-1-4, (Appendix E). The DD shall be approved (by signature) by the Responsible Manager, Technical Reviewer(s), QA Reviewer, and the SCM Coordinator following the process described in Appendix A. The design may necessitate the modification of the RD and VVP.

Note: There may be more than one design document (which may be combined into one document) created during software development. For example a high-level design may be developed to match the code design to the requirements, and define the overall architecture of the code (define modules and subroutines and their purpose, define data structures, define what routine calls what routine, etc.). Another detailed design document may be developed to define how the modules will function in detail, define call interfaces between routines, defines data types, etc. A detailed design as its name implies, is very detailed down to level of almost writing the code (pseudocode).

2.3.4 Implementation Phase

The following documents are produced during this phase:

- The Implementation Document (ID) - provides the source code listing or location where the source code is stored in a configuration management tool and the process of generating executable software, and
- The User's Manual (UM) - provides information to assist users in understanding and using the software.

The design as described in the DD is used as the basis for the software development, and may need to be modified to reflect changes identified in the implementation phase.

The Code Team/Sponsor shall develop the ID and UM using the phase criteria listed on the Implementation Document Criteria Form NP 19-1-5, (Appendix F) and the User Manual Criteria Form NP 19-1-6 (Appendix G). The ID and UM shall be approved (by signature) by the Responsible Manager, Technical Reviewer(s), QA Reviewer and the SCM Coordinator following the process described in Appendix A.

2.3.5 Validation Phase

The Validation Document (VD), produced during this phase, documents the test case input and output files, and the evaluation of the results versus the acceptance criteria identified in the approved VVP for each test case.

The validation phase consists of executing and reviewing the test cases identified in the approved VVP to demonstrate that the developed software meets the requirements defined for it in the RD. The Code Team/Sponsor shall develop and approve the Validation Document using the phase criteria listed on the Validation Document Criteria Form NP 19-1-7, (Appendix H). The VD shall be approved (by signature) by the Responsible Manager, Technical Reviewer(s), QA Reviewer and the SCM Coordinator following the process described in Appendix A.

2.3.6 Installation and Checkout Phase

The following documents are produced during this phase:

- The Installation and Checkout (I&C) Form NP 19-1-8 (Section 2.3.6.1 and Appendix I)
- The Access Control Memorandum (Section 2.3.6.2) and
- The Approved Users Memorandum (Section 2.3.6.3).

2.3.6.1 The Installation and Checkout Form

The I&C Form (produced by the Code Team/Sponsor) provides evidence of:

- The installation of the baseline software on the production computer including re-compiling and linking if necessary,
- The execution of the validation cases on the production computer (installation on a network of identical computers running identical operating systems requires testing on only one of the machines), and the testing with selected test cases (those identified as appropriate for installation and checkout) from the approved VVP to demonstrate acceptable performance on the target computer. The results of the testing must be attached to the I&C Form in a regression test report or in a separate regression test report that is referenced on the I&C Form. If the testing is documented in a separate regression test report, that document must go through the same review process described in Appendix A. If testing for the Validation Phase in Section 2.3.5 was performed on the production computer, then the test cases need not be rerun (provide reference to the VD).

The I&C Form shall be approved (by signature) by the Responsible Manager, Technical Reviewer, QA Reviewer and the SCM Coordinator.

2.3.6.2 Access Control Memorandum

The Access Control Memorandum establishes, to the extent appropriate, controls to permit authorized and prevent unauthorized access of the software.

The Code Team/Sponsor shall document access control measures in the Access Control Memorandum. When specifying access control on a system-wide basis, document or provide reference to the Access Control Memorandum describing system specific controls.

Note: Access Control Memorandum may be changed without modification of the Software Installation and Checkout Form NP 19-1-8 (Appendix I).

2.3.6.3 Approved Users Memorandum

The Approved Users Memorandum (produced by the Code Team/Sponsor or manager) identifies users for a particular code. Users may be identified by name, department, or group, etc. The Approved Users Memorandum shall be included or referenced as part of Installation and Checkout Phase Documentation.

Note: User list may be changed without modification of the Software Installation and Checkout Form NP 19-1-8 (Appendix I).

2.3.7 Maintenance Phase

This section provides the process for requesting, controlling, and implementing changes to software configuration baselines. Changes to software production baselines shall be formally evaluated,

approved or disapproved, and the change appropriately reflected in associated baseline documentation.

2.3.7.1 Production Software and/or Baseline Document Change Control

When necessary, the Code Team/Sponsor shall propose changes to the software baseline, using the Change Control Form, Form NP 19-1-9, (Appendix J), which shall be approved (by signature) by the Responsible Manager, Technical Reviewer(s), QA Reviewer and the SCM Coordinator. Instructions on completing this form can be found in Appendix J.

There are two types of proposed changes:

- Major changes – include new requirements, new design, new models, new implementation, require a new baseline (i.e., SQAP, RD, DD, VVP, ID, UM, VD) to be documented. In addition to revising every baseline document, a Change Control Form, and the Installation and Checkout Form are used.

- Minor changes – do not affect the requirements or design and can be documented with addenda (no more than three addenda's per baseline document) or page changes to the affected baseline document, in addition to the Change Control Form and the Installation and Checkout Form.

Once the changes to the Software Baseline are approved the SCM Coordinator shall:

- identify affected software configuration baselines.
- verify unique revision identifier.
- inform affected users of approved changes.
- redline/update baseline list.
- forward the Change Control Form to the SNL WIPP Records Center.

Once the changes to the Software Baseline are approved the Code Team/Sponsor shall:

- Perform modifications to software and/or associated baseline documentation identified in the Change Control Form in accordance with the appropriate sub-sections of this procedure. The version of the revision(s) should reflect the nature and scope of the change (see Section 2.0).
- Ensure that all baseline versions are consistent (see Section 2.0).
- Re-compile the software (if the modifications require this to be done). The software shall be subject to regression testing to: detect errors introduced during the modification of the systems or system components, verify that the modifications have not caused unintended adverse effects, and verify that the modified systems or system components still meet specified requirements.
- Document per the Installation and Checkout phase of Section 2.3.6. The degree of software validation shall be reasonable and commensurate with the nature and scope of the change.

2.3.7.1.1 Coding Documentation Standards

Changes to software may be accompanied by documentation within the code describing the change, the date the change was made, and the name of the person responsible for implementing the change. This documentation may be placed in the vicinity of the change, as well as at the top of the code prior to the first executable line. This is done as a best practice.

2.3.7.2 System Software and Hardware Changes

The Code Team/Sponsor or System Administrator shall propose system software or hardware changes that may have an impact to the results generated by production software or cause recompilation of production software, using the Change Control Form NP 19-1-9 (Appendix J).

Examples of changes to system software or hardware:

- changes to the operating system such that the version or level identifier changes
- changes to the Central Processing Unit (CPU)
- database management system change

The Code Team/Sponsor or System Administrator shall:

- initiate the approved system modification to the system software and/or hardware.
- initiate regression testing when significant system software or hardware changes are made (after changes have been performed on the production computer and prior to the next use of the baseline software) on all affected production baseline software in accordance with Section 2.3.6, Installation and Checkout.

Note: Small increments in the version of system software or hardware because of patches are not considered significant and do not require regression testing. A significant change would be when the release version changes the major number of the version, e.g. Solaris 11.x to 12.x.

2.3.7.3 Regression Testing

Each production baseline software code has a Verification and Validation Plan (VVP) which includes acceptance criteria each test case must meet. The Code Team/Sponsor will compare the results of the regression test to previous test results to ensure the test cases meet their acceptance criteria and prepare a Regression Test Report (see Appendix M).

There may be differences between the output files of the regression tests and those of the previous testing. The following types of differences are most common. These differences are expected and are completely acceptable:

- Differences due to code executable names.
- Differences due to code version numbers.
- Differences due to code build dates and times.
- Differences due to run dates and times.
- Differences due to different file names.
- Differences due to different directory and class names.
- Differences due to different user names.
- Differences due to platform and system version.
- Differences due to different execution statistics.

In addition, there may be minor numerical differences between the output files. Numerical differences will be examined by the Code Team/Sponsor to determine the origin of the differences and whether the differences affect the code's performance. Numerical differences may be determined to be acceptable based on the analysis of each difference. If any code does not meet the acceptance criteria it fails regression testing and it will undergo full validation testing after the problem has been resolved.

After the Regression Test Report has been reviewed and approved, per NP 6-1, *Document Review Process*, and the code has been validated for use on a particular platform, Code Team/Sponsor will

complete an Installation and Checkout Form. The Code Team/Sponsor may attach the completed Regression Test Report to the Installation and Checkout Form or may make reference to the Regression Test Report for that code.

2.3.7.4 Software Problem Report (SPR)

Whenever a software problem is identified, the Code Team/Sponsor shall evaluate the problem to determine if it is indeed a problem (as opposed to user error). If it is a problem, the SPR process shall be followed.

The Code Team/Sponsor shall classify the problem as major if it could significantly impact previous uses of code or if it will require significant modification to the software; otherwise classify it as minor.

The Code Team/Sponsor shall complete the Software Problem Report Form, NP 19-1-10, and forward it to the Responsible Manager for concurrence on classification (i.e., major, minor). The Code Team/Sponsor shall describe the software problem in field 3 of the Software Problem Report Form (attach pages as needed). The Code Team/Sponsor may also recommend actions to resolve the problem in field 3 of the SPR.

For *major problems,* the Responsible Manager shall identify affected users to be notified of the problem and designate qualified personnel to identify and evaluate the impact of the software problem. The impact analysis should describe the impact to any analysis, which used the output, produced by the subject software version. If additional calculations are needed or the analysis is to be redone, NP 9-1 shall be followed for any changes. If there was no impact, provide justification for using the analysis “as is”. If the problem is a condition adverse to quality, a CAR shall be initiated per NP 16-1.

For *minor problems,* the impact analysis can be performed by the Code Team/Sponsor.

The SPR form is then forwarded to the appropriate staff for concurrence and signatures.

The SCM Coordinator shall assign an SPR number to the Software Problem Report. The SCM Coordinator shall also update/redline the Software Baseline List. The SCM Coordinator shall forward the SPR Form to the SNL WIPP Records Center.

If necessary, the Code Team/Sponsor shall propose changes to correct the applicable baseline components per Section 2.3.7.1.

After the software problem has been corrected, the Code Team/Sponsor shall complete the Software Problem Closure Report Form (SPCR), NP 19-1-11, and forward it to the appropriate staff for concurrence and signatures to complete the correction process and to the QA Staff for verification. The SCM Coordinator shall forward the SPRC Form to the SNL WIPP Records Center. The letter “c” will be placed after the SPR number to flag the SPR as closed on the Software Baseline List.

2.3.7.5 Configuration Management (Configuration Identification and Status Accounting)

This section provides the process for defining the configuration of software products, establishing software configuration baselines, and tracking the status of baseline changes. A software configuration baseline consists of the source code and baseline documents, providing objective evidence of technical adequacy.

The SCM Coordinator shall maintain a Software Baseline List, and make it available upon request. The SCM Coordinator performs a completeness review to ensure that necessary components of configuration management are present.

For **Compliance Decision**, the Software Baseline List shall contain:

- code name and version,
- code version date,
- Code Team/Sponsor name,
- code classification (acquired or developed),
- SQAP version,
- RD version,
- VVP version,
- DD version,
- ID version,
- UM version,
- VD version,
- list of approved users (may be listed by name, department, or group, etc.)
- list of approved system software/hardware configurations,
- list of Software Problem Report (SPR) numbers (see Section 2.3.7.3), and
- status of approved changes which are in process.
- I&C date

The SCM coordinator shall redline the Software Baseline List when new or revised software products and/or documentation baselines are approved for use. A redlined list shall be maintained until a new baseline list is issued. The SCM coordinator shall periodically (at least once every calendar year), issue the baseline software list identifying all software with no approved users as candidates for retirement.

The Code Team/Sponsors shall review the Software Baseline List for accuracy and for codes that may be retired from production use. (Code retirement is addressed in Section 2.3.8). Code Team/Sponsors shall report any changes or inaccuracies to the SCM Coordinator.

2.3.8 Retirement Phase

To retire a code, the Code Team/Sponsor or Manager issues a memorandum to the SCM Coordinator requesting that the code be retired, and provide a reason for the retirement.

The SCM Coordinator marks the code as retired in the baseline software list.

The System Administrator and/or Code Team/Sponsor shall take action to prevent the use of the retired code. This could involve removal of the software from the computer or the changing of execution privileges.

To bring a code out of retirement, to be used and placed back on the baseline software list, an analysis needs to be performed by the Code Team/Sponsor to determine if the code and its documentation are still valid. If the documentation is still correct, the code must be tested to ensure it still functions as stated in the Validation Document. An Installation and Checkout Form must be filled out to complete the process for having it added back as an approved code (per section 2.3.6).

2.4 Interim Use of Unqualified Software to Support Programmatic Decisions

With written permission granted in advance by the Sandia Carlsbad Senior Manager, relying on input from the Responsible Manager, and Quality Assurance, some software that is required to support various Analysis Reports may need to be used prior to full qualification.

Software covered by this section is not to be used for any other purposes or any other milestone deliverables and its applicability shall be limited to Programmatic Decisions. This section describes the requirements and process methodologies that will permit the interim use and controls of unqualified software in products that are currently being developed to support the SNL WIPP program.

2.4.1 Code Team/Sponsor:

Determine the need to use unqualified software based on the work scope, deliverable schedule, and complexity of confirmation once qualified.

Prepare an Analysis Plan (AP) in accordance with NP 9-1 “Analyses”. The AP, in this case, shall outline how the unqualified software will be used, a schedule for qualification, and comparison confirmation methodologies, including acceptance criteria to be used to determine the extent of impact evaluations that may be applicable once the software is qualified.

2.4.2 Sandia Carlsbad Senior Manager or designee:

The Sandia Carlsbad Senior Manager, or designee will approve the use of unqualified software to support programmatic decisions by signing the associated AP. This signature serves as the written permission, required in Section 2.4.

2.4.3 SCM Coordinator:

Establish and maintain an unqualified software list containing the code name and version, version date, System Configuration, Code Team/Sponsor, and Code Classification.

2.4.4 Code Team/Sponsor:

a) Install the unqualified software in accordance with Section 2.3.6 Installation and Checkout Phase and submit an Implementation Document per Section 2.3.4 for the Implementation phase. Initiate a CAR in accordance with NP 16-1 to track the use of the data generated with the unqualified software.
b) Continue work on the documentation and qualification aspects of the software in accordance with this procedure.
c) Once the software has been qualified and baselined in accordance with this procedure, compare the test cases run on the qualified version with each of the test cases run on the unqualified software versions that were used to generate data, develop data or output.
    1) If the comparison indicates that no differences exist or that the differences can be justified, all previous data generated from that version of software are acceptable. Justification for the differences must be documented.
    2) If differences exist that cannot be justified, all previous data generated must be re-run, using the qualified version of the software.
    3) Once the software has been qualified and baselined and the impact reviews have been resolved, submit the record copy to the SCM Coordinator for inclusion in the software records package.

If the software will not be used in a production environment then retire the software per Section 2.3.8 of this procedure.

3.0 Records

The following records, generated through implementation of this procedure, shall be prepared and submitted to the SNL WIPP Records Center in accordance with NP 17-1 (Records):

QA Record

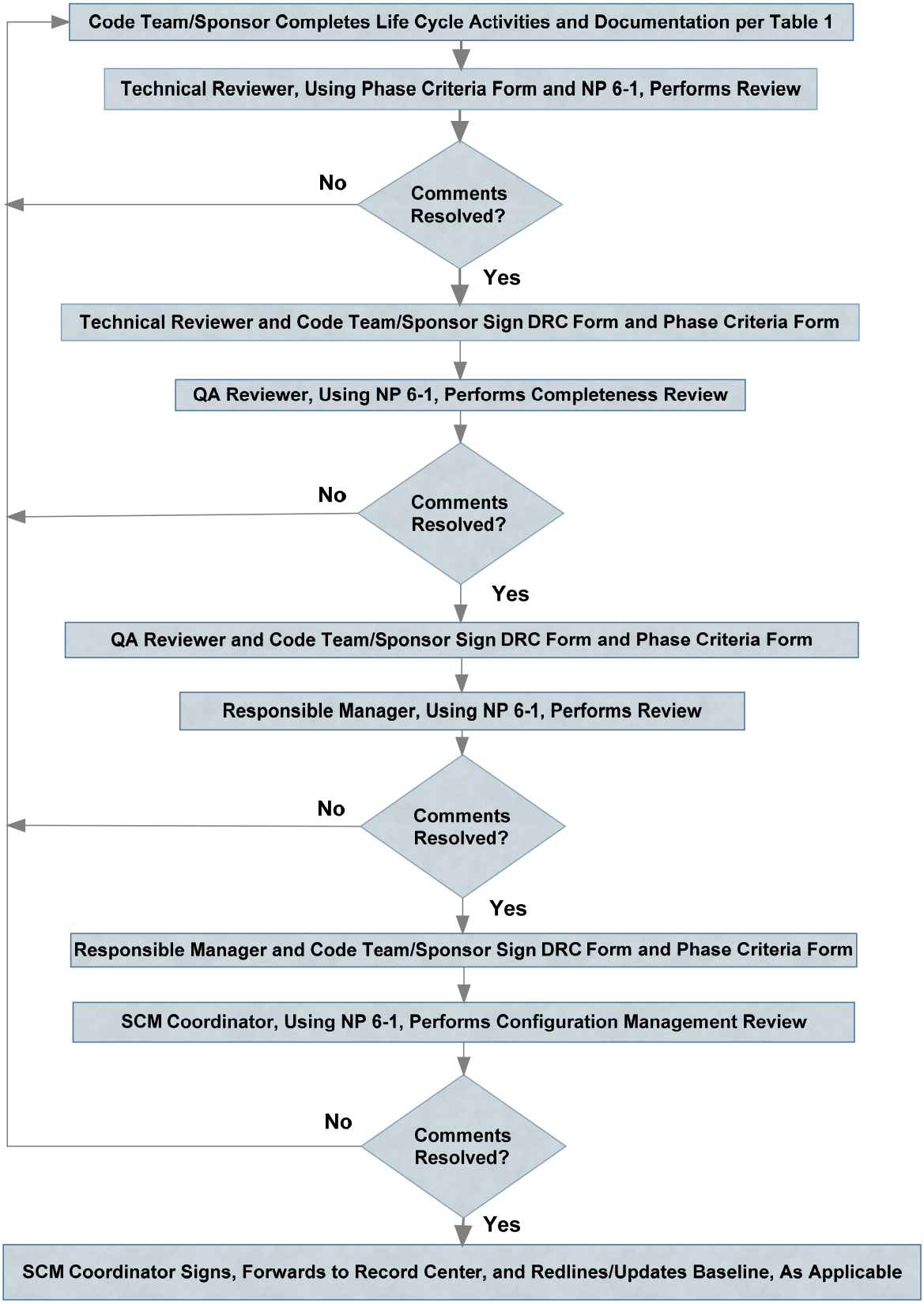
- Software Baseline List
- Software QA Plan
- Requirements Document (RD)
- Verification and Validation Plan (VVP)
- Design Document (DD)
- User's Manual (UM)
- Implementation Document (ID) (the source code may be stored in a configuration management tool in lieu of records)
- Validation Document (VD)
- Software QA Plan Criteria (NP 19-1-1)
- Requirements Document Criteria (NP 19-1-2)
- Verification and Validation Plan Criteria (NP 19-1-3)
- Design Document Criteria (NP 19-1-4)
- Implementation Document Criteria (NP 19-1-5)
- User Manual Criteria (NP 19-1-6)
- Validation Document Criteria (NP 19-1-7)
- Software Installation and Checkout (NP 19-1-8)
- Change Control (NP 19-1-9)
- Software Problem Report (NP 19-1-10)
- Software Problem Closure Report (NP 19-1-11)
- Access Control Memorandum
- Approved User Change Memorandum
- Code Retirement Request Memorandum
- Regression Test Report

4.0 Appendices

Appendix A: Software Life-Cycle Process Flow Chart
Appendix B: Form NP 19-1-1, Software QA Plan Criteria
Appendix C: Form NP 19-1-2, Requirements Document Criteria
Appendix D: Form NP 19-1-3, Verification and Validation Plan Criteria
Appendix E: Form NP 19-1-4, Design Document Criteria
Appendix F: Form NP 19-1-5, Implementation Document Criteria
Appendix G: Form NP 19-1-6, User's Manual Criteria
Appendix H: Form NP 19-1-7, Validation Document Criteria
Appendix I: Form NP 19-1-8, Software Installation and Checkout
Appendix J: Form NP 19-1-9, Change Control
Appendix K: Form NP 19-1-10, Software Problem Report (SPR)
Appendix L: Form NP 19-1-11, Software Problem Closure Report (SPCR)
Appendix M: Regression Test Report Format

Key for check boxes in all forms where applicable:
Check **Yes** for each item reviewed and found acceptable
Check **N/A** for items not applicable, where applicable as based on code functionality

Appendix A
Software Life-Cycle Process Flow Chart



Appendix B

NUCLEAR WASTE MANAGEMENT PROCEDURE Sandia National Laboratories	**Software QA Plan Criteria**	**Form Number:** NP 19-1-1 **Page 1 of 1**

1. **Software Name:** _____
2. **Software Version:** _____
3. **Document Version:** _____
4. **ERMS #:** _____

Prior to sign-off of the SQA Plan, all items shall be appropriately addressed by the code sponsor so that "**Yes**" or "**N/A**" may be checked. Include this form as part of the SQA Plan.

5. **Software Identification:** Are software name, version and scope identified answering why we are doing this and what problem will be solved?	☐ Yes	
6. **Deviations:** If there are deviations from the Lifecycle required documentation, is the deviation adequately explained and is it appropriate?	☐ Yes	☐ N/A
7. **Documents:** Are the documents to be prepared, reviewed and maintained identified?	☐ Yes	
8. **Organizations:** Are the organizations responsible for work quality assurance identified with tasks (a schedule for qualification) and responsibilities?	☐ Yes	
9. **Development Methods:** Are the standards, conventions, techniques methods and procedures (SOPs, WPs etc.) identified for use in establishing and maintaining integrity of code, embodied mathematic models, [illegible], and processes?	☐ Yes	
10. **Problem Reporting:** Is there a process for documenting and reporting software discrepancies, evaluating the impact of errors on previous [illegible]ions, and determining the appropriate corrective action(s)?	☐ Yes	
11. **External Interfaces:** Are required interactions with people, hardware, and other software identified?	☐ Yes	
12. **Completeness:** Is the plan complete?	☐ Yes	
13. **Verifiability:** Can meeting the plan be verified?	☐ Yes	
14. **Consistency:** Is the plan consistent internally and with other software?	☐ Yes	
15. **Technical Feasibility:** Is the plan technically feasible and can it result in a useable code?	☐ Yes	

16.	**Code Team/Sponsor's Name** (*print*)	Signature	Date
17.	**Technical Reviewer's Name** (*print*)	Signature	Date
18.	**QA Reviewer's Name** (*print*)	Signature	Date
19.	**Responsible Manager's Name** (*print*)	Signature	Date
20.	**SCM Coordinator's Name** (*print*)	Signature	Date

Sample Form

Appendix C

NUCLEAR WASTE MANAGEMENT PROCEDURE Sandia National Laboratories	**Requirements Document Criteria**	**Form Number:** NP 19-1-2 **Page 1 of 1**

1. **Software Name:** _____
2. **Software Version:** _____
3. **Document Version:** _____
4. **ERMS #:** _____

Prior to sign-off of the RD, all items shall be appropriately addressed by the code sponsor so that "**Yes**" or "**N/A**" may be checked. Include this form as part of the RD.

5. **Functionality:** Are the functions that the software is to perform adequately identified? ☐ Yes
6. **Performance:** Are time-related software operations issues, e.g., speed, recovery time, or response time identified, where applicable as based on the code functionality? ☐ Yes ☐ N/A
7. **Design Constraints:** Are elements that will restrict design options identified? ☐ Yes ☐ N/A
8. **Attributes (non-time-related):** Are the following identified, where applicable as based on the code functionality:
   - portability? ☐ Yes ☐ N/A
   - acceptance criteria? ☐ Yes
   - maintainability? ☐ Yes ☐ N/A
9. **External Interfaces:** Are the following interfaces identified, where applicable as based on the code functionality:
   - People? ☐ Yes ☐ N/A
   - Hardware? ☐ Yes ☐ N/A
   - Software? ☐ Yes ☐ N/A
10. **Completeness:** Are the requirements complete? ☐ Yes
11. **Verifiability:** Can meeting the requirements be verified? ☐ Yes
12. **Consistency:** Are requirements consistent with each other? ☐ Yes
13. **Technical Feasibility** Are the requirements technically feasible and can they result in a useable code? ☐ Yes

14. _____ **Code Team/Sponsor's Name** (*print*) _____ Signature _____ Date
15. _____ **Technical Reviewer's Name** (*print*) _____ Signature _____ Date
16. _____ **QA Reviewer's Name** (*print*) _____ Signature _____ Date
17. _____ **Responsible Manager's Name** (*print*) _____ Signature _____ Date
18. _____ **SCM Coordinator's Name** (*print*) _____ Signature _____ Date

Sample Form

Form NP 19-1-2 Instructions

1 – 4. These fields are needed for configuration management. Please supply the software name and version for which the RD is being written. Provide the RD Document Version. Follow Version requirements listed in Section 2.0.

5. Functionality. Functional requirements define what the software product must accomplish.

6. Performance. Clearly describe all required time performance issues.

7. Design Constraints. Clearly describe any functional requirements that will later restrict design options.

8. Attributes.

   - Portability. Describe any requirements for using the code on more than one platform
   - Acceptance Criteria. Acceptable result for a given functional requirement typically includes a quantification of acceptable error range per %. Acceptance criteria specifies the outputs and features required to demonstrate acceptable performance and provides a qualitative basis for each required output or feature to be evaluated
   - Maintainability. The structure and style of the requirement allows that necessary changes can be made.

9. External Interfaces. Describe any interfaces with users that will be functional requirements (GUI interfaces for example).

10. Completeness. Each requirement describes a result that must be achieved. All requirements together describe all functionality that the software product will provide.

11. Verifiability. Functional requirements must be implementable as source code.

12. Consistency. Individual requirements are not in conflict with each other.

13. Technical Feasibility. The requirements can be implemented under existing constraints.

Sample Form

Appendix D

NUCLEAR WASTE MANAGEMENT PROCEDURE Sandia National Laboratories	**Verification and Validation Plan Criteria**	**Form Number:** NP 19-1-3 **Page 1 of 1**

1. **Software Name:** _____
2. **Software Version:** _____
3. **Document Version:** _____
4. **ERMS #:** _____

Prior to sign-off of the VVP, all items shall be appropriately addressed by the code sponsor so that "**Yes**" or "**N/A**" may be checked. Include this form as part of the VVP.

5. **Sufficient Test Cases** ☐ Yes
   Does the VVP identify sufficient test cases and acceptance criteria to ensure the final software and end product satisfies the requirements of the RD? (Check Yes if peer review is identified to fulfill the validation requirements)
6. **Adequacy of Test Cases** ☐ Yes
   Do the test cases demonstrate that the code adequately performs all intended functions and produces valid results for problems encompassing the range of permitted usage?
7. **Operational Control** ☐ Yes ☐ N/A
   If the software is used for operational control, do the tests demonstrate required performance over the range of operation of the controlled function or process?
8. **Unintended Functions** ☐ Yes ☐ N/A
   Do the test cases show that the code does not perform any unintended function that either by itself or in combination with other functions can degrade intended outcomes of the software?
9. **Test Result Validation.** (check one or more, where applicable based on code functionality)
   The test results will be compared to the following:
   - hand calculations, ☐ Yes ☐ N/A
   - manual inspection, ☐ Yes ☐ N/A
   - calculations using comparable proven products, ☐ Yes ☐ N/A
   - empirical data and information from confirmed published data ☐ Yes ☐ N/A
   - and correlations and/or technical literature, ☐ Yes ☐ N/A
   - other validated software of similar purpose, ☐ Yes ☐ N/A
   - or independent software of similar purpose. ☐ Yes ☐ N/A

   A documented peer review will be performed. ☐ Yes ☐ N/A
   Do the test cases describe how the code results will be validated? ☐ Yes
10. **Does the VVP specify the following, where applicable as based on code functionality?**
    (a) required tests and test sequence ☐ Yes ☐ N/A
    (b) required ranges of input parameters ☐ Yes ☐ N/A
    (c) identification of the stages at which testing is required ☐ Yes ☐ N/A
    (d) criteria for establishing test cases ☐ Yes ☐ N/A
    (e) requirements for testing logic branches ☐ Yes ☐ N/A
    (f) requirements for hardware integration ☐ Yes ☐ N/A
    (g) anticipated output values ☐ Yes ☐ N/A
    (h) acceptance criteria ☐ Yes ☐ N/A
11. **Installation and Regression Testing**
    Are test cases which are suitable for installation testing and regression testing identified in the set of verification and validation test cases? ☐ Yes

12.	**Code Team/Sponsor's Name** (*print*)	Signature	Date
13.	**Technical Reviewer's Name** (*print*)	Signature	Date
14.	**QA Reviewer's Name** (*print*)	Signature	Date
15.	**Responsible Manager's Name** (*print*)	Signature	Date
16.	**SCM Coordinator's Name** (*print*)	Signature	Date

Sample Form

Appendix E

NUCLEAR WASTE MANAGEMENT PROCEDURE Sandia National Laboratories	**Design Document Criteria**	**Form Number:** NP 19-1-4 **Page 1 of 1**

1. **Software Name:** _____
2. **Software Version:** _____
3. **Document Version:** _____
4. **ERMS #:** _____

Prior to sign-off of the DD, all items shall be appropriately addressed by the code sponsor so that "**Yes**" may be checked. Include this form as part of the DD.

Are the following appropriately defined and documented in the DD?

5. Major Software Components ☐ Yes
6. Technical description of the software with respect to: theoretical basis, embodied mathematical model, major control flow, control logic, and data structures ☐ Yes
7. Functionalities and interfaces of object components, functions, and subroutines. ☐ Yes
8. Allowable or prescribed ranges of inputs and outputs ☐ Yes
9. Verifiability: Is the design verifiable through testing or other means? ☐ Yes
10. Consistency and Traceability: Is the design consistent with and traceable to the software's requirements? ☐ Yes
11. Technical Feasibility: Is the design technically feasible? ☐ Yes
12. Implementation: Is the design presented in sufficient detail to allow for implementation as computer software? ☐ Yes

13.	_____	_____	_____
	Code Team/Sponsor (*print*)	Signature	Date
14.	_____	_____	_____
	Technical Reviewer (*print*)	Signature	Date
15.	_____	_____	_____
	QA Reviewer (*print*)	Signature	Date
16.	_____	_____	_____
	Responsible Manager (*print*)	Signature	Date
17.	_____	_____	_____
	SCM Coordinator (*print*)	Signature	Date

Sample Form

Appendix F

NUCLEAR WASTE MANAGEMENT PROCEDURE Sandia National Laboratories	**Implementation Document Criteria**	**Form Number:** NP 19-1-5 **Page 1 of 1**

1. **Software Name:** _____
2. **Software Version:** _____
3. **Document Version:** _____
4. **ERMS #:** _____

Prior to sign-off of the ID, all items shall be appropriately addressed by the code sponsor so that "**Yes**" or "**N/A**" may be checked. Include this form as part of the ID.

5. **Source Code** ☐ Yes ☐ N/A

   Is the source code or a reference to its location in a configuration management tool provided?

   Note: If the source code is not controlled in a configuration management tool then a hardcopy of the source is required. (*Check "N/A" for commercially obtained software for which source code was not provided.*)

6. **Coding Standards** ☐ Yes ☐ N/A

   Are the coding standards and conventions which were adhered to in the development of the software identified?

7. **Coding Standards Implementation** ☐ Yes ☐ N/A

   Does the source code adhere to the coding standards and conventions defined in the ID?

8. **Executable Generation** ☐ Yes ☐ N/A

   Was the executable generation process documented?

9. **Object Libraries** ☐ Yes ☐ N/A

   Were object libraries used?

   If object libraries were used, are their names and locations provided?

10. **Implementation Requirements** ☐ Yes ☐ N/A

    Was the code implemented according to the requirements of the RD and where applicable the DD?

11.	**Code Team/Sponsor's Name** (*print*)	Signature	Date
12.	**Technical Reviewer's Name** (*print*)	Signature	Date
13.	**QA Reviewer's Name** (*print*)	Signature	Date
14.	**Responsible Manager's Name** (*print*)	Signature	Date
15.	**SCM Coordinator's Name** (*print*)	Signature	Date

Sample Form

Appendix G

NUCLEAR WASTE MANAGEMENT PROCEDURE Sandia National Laboratories	**User's Manual Criteria**	**Form Number:** NP 19-1-6 **Page 1 of 1**

Does the user's manual contain as appropriate:

1. **Software Name:** _____
2. **Software Version:** _____
3. **Document Version:** _____
4. **ERMS #:** _____

Prior to sign-off of the User's Manual, all items shall be appropriately addressed by the code sp[illegible] th[illegible] "**Yes**" or "**N/A**" may be checked. Include this form as part of the User's Manual.

5. A statement(s) of functional requirements (consistent with those in the RD) and system limitations? ☐ Yes
6. An explanation of the mathematical model and numerical models, where applicable as based on code func[illegible]y? ☐ Yes ☐ N/A
7. Physical and mathematical assumptio[illegible] whe[illegible]licable as based on code functionality? ☐ Yes ☐ N/A
8. The capabilities and limitations in[illegible]ent in[illegible]he [illegible]are? ☐ Yes
9. Instructions that describe the u[illegible]'s i[illegible]erac[illegible]n w[illegible] software? ☐ Yes
10. The identificatio[illegible]ut p[illegible]rame[illegible]s [illegible]mats, and valid ranges? ☐ Yes
11. Messa[illegible]itiate[illegible]as [illegible]esu[illegible] of im[illegible]oper input and how [illegible]e use[illegible]car[illegible]espo[illegible]d? ☐ Yes
12. [illegible]iden[illegible]cat[illegible]n and [illegible]escription of output specifications and formats? ☐ Yes
13. A de[illegible]cript[illegible] any required training necessary to use the software? ☐ Yes
14. [illegible]entification of components of the code that were not tested? ☐ Yes ☐ N/A

15. _____ **Code Team/Sponsor** (*print*) _____ Signature _____ Date
16. _____ **Technical Reviewer** (*print*) _____ Signature _____ Date
17. _____ **QA Reviewer** (*print*) _____ Signature _____ Date
18. _____ **Responsible Manager** (*print*) _____ Signature _____ Date
19. _____ **SCM Coordinator** (*print*) _____ Signature _____ Date

Sample Form

Appendix H

NUCLEAR WASTE MANAGEMENT PROCEDURE Sandia National Laboratories	**Validation Document Criteria**	**Form Number:** NP 19-1-7 **Page 1 of 1**

1. **Software Name:** _____
2. **Software Version:** _____
3. **Document Version:** _____
4. **ERMS #:** _____

Prior to sign-off of the VD, all items shall be appropriately addressed by the code sponsor so that "**Yes**" or "**N/A**" may be checked. Include this form as part of the VD.

5. **Is the following information included, where applicable?**
   - (a) computer program and version tested ☐ Yes
   - (b) computer hardware and operating system used ☐ Yes
   - (c) test equipment and calibrations ☐ Yes ☐ N/A
   - (d) date of test ☐ Yes
   - (e) tester or data recorder ☐ Yes
   - (f) simulation models used, ☐ Yes ☐ N/A
   - (g) test problem input and output files ☐ Yes
   - (h) results and acceptability ☐ Yes
   - (I) action taken in connection with any deviations noted ☐ Yes ☐ N/A
6. **Test Result Validation**
   The test results were compared to the following (check one or more, where applicable as based on code functionality):
   - hand calculations, ☐ Yes ☐ N/A
   - manual inspection, ☐ Yes ☐ N/A
   - calculations using comparable proven problem, ☐ Yes ☐ N/A
   - empirical data & data from confirmed published data and correlations and technical literature, ☐ Yes ☐ N/A
   - other validated software of similar purpose, ☐ Yes ☐ N/A
   - other independent software of similar purpose. ☐ Yes ☐ N/A
7. **Test Documentation Acceptability** ☐ Yes
   Do the test results meet the acceptance criteria identified in the approved VVP?
8. **Test Documentation Repeatability** ☐ Yes
   Are the tests documented in sufficient detail such that they can be repeated?
9. **Computer File Documentation** ☐ Yes
   Are the test case input and output files included in the Validation Document or a reference to their location in a configuration management tool provided?
10. **Understandability of Documentation** ☐ Yes
    Are the validation methods, test data, results, and conclusions documented in a form that can be understood by an independent, technically competent individual?

11.	**Code Team/Sponsor** (*print*)	Signature	Date
12.	**Technical Reviewer** (*print*)	Signature	Date
13.	**QA Reviewer** (*print*)	Signature	Date
14.	**Responsible Manager** (*print*)	Signature	Date
15.	**SCM Coordinator** (*print*)	Signature	Date

Sample Form

Form NP 19-1-7 Instructions

The Code Team/Sponsor or designee (e.g. tester) shall execute the test cases and compare results to the acceptance criteria identified in the approved VVP. Any tests performed during the implementation phase which were not previously documented and reviewed should be formally documented, as appropriate, and the VVP revised to reflect the additional tests.

"Manual Inspection" in Item 6 refers to manual activities which do not involve numerical manipulations. These include visual inspection of table reformatting or plotting, and concurrence of qualitative acceptance criteria such as trends in results due to input parameter variations.

In order to allow for comparison of test results to other independent software of similar purpose, the following criteria must be met:

- comparison of test results to any of the four previously listed methods in [illegible] is impossible or impractical;
- the computer codes were independently developed. This must mean development by different individuals. This should include the use of different theoretical bases, use of different modeling strategies, or different mathematical models.
- validation of any theoretical basis or mathematical model which is not considered a conventional, generally accepted solution technique for that application must be performed via another method.

The tests should demonstrate the capability of the software to produce valid results for problems encompassing the range of permitted usages as defined by the User's Manual.

Sample Form

Appendix I

NUCLEAR WASTE MANAGEMENT PROCEDURE Sandia National Laboratories	**Software Installation and Checkout**	**Form Number:** NP 19-1-8 **Page 1 of 1**

General Information

1. Software Name: _____
2. Software Version: _____
3. ERMS #: _____
4. Code Classification: _____
   a. ID Document ERMS#: _____
   b. VD Document ERMS# (to which the test cases are compared): _____

Executable or Object Information

5. Executable or Object Name (*include path*): _____
6. Executable or Object Size (*bytes*): _____
7. Executable or Object Date: _____

Compilation Information

8. Hardware System: _____
9. Operating System: _____

Installation and Checkout Information

10. Hardware System: _____
11. Operating System: _____
12. Any SPRs outstanding? ☐ Yes ☐ No SPR No(s): _____
13. Any SPRs closed as a result of this Installation and Checkout? ☐ Yes ☐ No SPR No(s): _____

Software Information

14. Directory/Path ERMS Library: _____
15. Procedure(s): _____
16. Utilities: _____
17. Input Files: _____
18. Output Files: _____
19. Test Evaluation:
    Test results fully met specified acceptance criteria ☐ Yes
    ☐ attached, ☐ referenced, ERMS# _____
20. Access Control and Approved User Memo are attached to the I&C or are referenced:
    ☐ attached, ☐ referenced, ERMS# _____

21.	_____	_____	_____
	Code Team/Sponsor (*print*)	Signature	Date
22.	_____	_____	_____
	Technical Reviewer (*print*)	Signature	Date
23.	_____	_____	_____
	QA Reviewer (*print*)	Signature	Date
24.	_____	_____	_____
	Responsible Manager (*print*)	Signature	Date
25.	_____	_____	_____
	SCM Coordinator (*print*)	Signature	Date

Sample Form

Appendix J

NUCLEAR WASTE MANAGEMENT PROCEDURE Sandia National Laboratories	**Change Control (Software/Hardware/Baseline Document)**	**Form Number:** NP 19-1-9 **Page** ____ **of** ____

1. **Software Name:** _____
2. **Software Version Identifier** a) Current: _____ b) Proposed: _____
3. **Software Classification** a) Current: _____ b) Proposed: _____
4. **ERMS #** _____
5. **Current Hardware/Software Platform:** _____
6. **Type of change:** ☐ Major ☐ Minor
7. **SPR No.(s):** (List SPR numbers to be resolved by the proposed change) _____
8. **SPR No.(s):** (List outstanding SPR numbers not being addressed by this change and a rationale for not resolving them) _____
9. **Proposed Changes:** (attach pages as needed or use continuation sheet at end of form)

Sample Form

	Document Affected	Required Resolution		
Software Quality Plan (SQAP)	☐ Yes ☐ No*	☐ Revision	☐ Page Change	☐ Addenda
Version No: _____	*Rationale _____			
New Version No: _____	_____			
Requirements Document (RD)	☐ Yes ☐ No*	☐ Revision	☐ Page Change	☐ Addenda
Version No: _____	*Rationale _____			
New Version No: _____	_____			
Verification and Validation Plan (VVP)	☐ Yes ☐ No*	☐ Revision	☐ Page Change	☐ Addenda
Version No: _____	*Rationale _____			
New Version No: _____	_____			
Design Document (DD)	☐ Yes ☐ No*	☐ Revision	☐ Page Change	☐ Addenda
Version No: _____	*Rationale _____			
New Version No: _____	_____			

Appendix J (continued)

Change Control (Software/Hardware/Baseline Document)	**Form Number: NP 19-1-9** **Page ____ of ____**

Validation Document (VD) ☐ Yes ☐ No* ☐ Revision ☐ Page Change ☐ Addenda

Version No: _____ *Rationale _____

New Version No: _____ _____

Implementation Document (ID) ☐ Yes ☐ No* ☐ Revision ☐ Page Change ☐ Addenda

Version No: _____ *Rationale _____

New Version No: _____ _____

User's Manual (UM) ☐ Yes ☐ No* ☐ Revision ☐ Page Change ☐ Addenda

Version No: _____ *Rationale _____

New Version No: _____ _____

10. Proposed System Software/Hardware change se

11. _____ _____ _____

Code Team/Sponsor's Name *(print)* Signature Date

12. _____ _____ _____

Technical Reviewer's Name *(print)* Signature Date

13. _____ _____ _____

QA Reviewer's Name *(print)* Signature Date

14. _____ _____ _____

Responsible Manager's Name *(print)* Signature Date

15. _____ _____ _____

SCM Coordinator's Name *(print)* Signature Date

Sample Form

Appendix J (continued)

Change Control (Software/Hardware/Baseline Document) **(continuation sheet)**	**Form Number: NP 19-1-9** **Page ____ of ____**

Continuation of Item 9, Proposed Changes:

Sample Form

Change Control Form Instructions (Form Number NP 19-1-9)

This form is for proposal and approval of changes to production baseline software, changes to software documentation, and/or changes to system software and hardware. Changes to system software and hardware applies to systems which are used by more than one person for running production baseline software.

General Instructions

For each entry listed, additional pages may be attached as needed.

1. Software Name: Enter the name of the software.

2. Software Version Identifier: On (a) enter the current software version identifier as listed on the Software Baseline Inventory List. On (b) enter the proposed version identifier.

3. Software Classification: On (a) enter the current classification (i.e., acquired, developed, or) on (b) enter the proposed classification.

4. ERMS # assigned to Change Control Form, obtained by SCM Coordinator.

5. Hardware/Software Platform: Enter the hardware platform on which the software currently resides and any applicable system software (required for the execution and use of the production baseline software).

6. Type of change: Select a box from this line to indicate whether changes are major or minor.

7. List the software problem report (SPR) number(s) if this change is to resolve any outstanding SPR(s).

8. List outstanding SPR number(s) not being addressed by this change and a rationale for not resolving them.

9. Proposed Changes: Use this section to describe the proposed changes to the code and the changes each document will be undergoing. For each document, list the current document version number (as it appears on the Baseline Inventory List) and (if applicable) the new document version number.

   Software QA Plan (SQAP): Describe any deviations to the Lifecycle document requirements.

   Requirements Document (RD): Describe any features that are being changed, added, or deleted. Describe if any requirements are moving from not tested to tested. Include a discussion of required test cases to demonstrate acceptable performance of new code features. Provide rational for regression testing if all existing test cases will not be rerun.

   Verification and Validation Plan (VVP): Describe test cases and acceptance criteria that are being changed, added, or deleted. Discuss how these test cases demonstrate that the code adequately performs all tested functions.

   Design Document (DD): Describe the extent of changes to the DD. Note how changes will be verifiable through testing or other means.

   Validation Document (VD): Describe if the VD will change to reflect changes to the VVP or will be updated for other reasons.

Sample Form

Implementation Document (ID): In general, all ID changes will be revisions, not addenda's. Describe what aspects of the coding will change User Manuals (UM): Describe what user instructions will be changed, added, or deleted.

10. System Software/Hardware Change Section
    Describe proposed changes to system software and/or hardware. Describe expected impact, if any, to production baseline software which resides on the system. Describe how changes to system software and/or hardware will be tested. Discuss what regression testing of baseline software will be required or describe why no regression testing of production baseline software will be needed. If testing is needed, it must address the change to the system to verify that the change has been installed properly and works properly.

11. Code Sponsor Signature.
    Code Sponsor signs for changes to baseline software.

12. Technical Reviewer Signature. Indicates concurrence with impact to baseline documentation.

13. QA Reviewer Signature. After signing form, QA Reviewer forwards to responsible manager.

14. Responsible Manager Signature. After signing form, RM forwards to SCM Coordinator.

15. SCM Coordinator Signature. SCM Coordinator assigns change control form or returns it to code sponsor for proper completion. After SCM coordinator signature, forward Change Control Form to the SNL WIPP Records Center.

Sample Form

Appendix K

NUCLEAR WASTE MANAGEMENT PROCEDURE Sandia National Laboratories	**Software Problem Report (SPR)**	**Form Number:** NP 19-1-10 **Page** ____ **of** ____

1. Software Name and Version: _____

2. SPR Classification:

☐ Major (Problems that cause the calculations to be re-run or necessitates a change to all baseline documents, if this is a condition adverse to quality initiate a CAR per NP 16-1. An impact statement is needed from each person designated by Responsible Manager)

☐ Minor (Everything else)

3. Summary of Error: [how to reproduce it and suggestions for fixing it (optional), attach pages as needed or use continuation sheet at end of form)

4. Impact Analysis (report reference with ERMS# and a decision to re-do or use the analysis as is) (attach pages as needed or use continuation sheet at end of form)

Title	ERMS#	Decision

5. _____ _____ _____
Code Team/Sponsor Name (*print*) Signature Date

6. _____ _____ _____
Technical Reviewer Name (*print*) Signature Date

7. _____ _____ _____
QA Reviewer Name (*print*) Signature Date

8. _____ _____ _____
Responsible Manager Name (*print*) Signature Date

9. _____ _____ _____
SCM Coordinator Name (*print*) Signature Date

10. SPR No. (Year and sequence, e.g., 04-001): _____

Sample Form

Appendix K (continued)

Software Problem Report (SPR) **(continuation sheet)**	**Form Number: NP 19-1-10** **Page ____ of ____**

Continuation of Item 3, Summary of Error, or Item 4, Impact Analysis:

Sample Form

Appendix L

NUCLEAR WASTE MANAGEMENT PROCEDURE Sandia National Laboratories	**Software Problem Closure Report (SPCR)**	**Form Number:** NP 19-1-11 **Page** ____ **of** ____

1. Software Name and Version: _____

2. SPR Number: _____

3. Summary of actions taken to resolve the SPR and ERMS Number of documentation where the resolution was documented.

4. List Software Name and Version for Code(s) which have no resolved problem identified by this SPR.

Software Name	Version

5. _____ _____ _____
Code Team/Sponsor Name (*print*) Signature Date

6. Qualification (List actions taken to verify completion of SPCR including all documentation verified)

7. _____ _____ _____
QA Reviewer Name (*print*) Signature Date

8. _____ _____ _____
Responsible Manager Name (*print*) Signature Date

9. _____ _____ _____
SCM Coordinator Name (*print*) Signature Date

Sample Form

Appendix M
Regression Test Report Format

Cover Page:
The cover page shall contain the following:

- Title
- Author's signature
- Technical Reviewer's signature
- QA Reviewer's signature
- Management Reviewer's signature
- SCM Coordinator's signature
- File Code (as necessary)

Summary and Conclusions:
The Summary and Conclusion section will provide a brief description of the code being regression tested, why the testing was necessary, the code version to which these results were compared (including platform and operating system information) and a discussion of the results of the comparison. State clearly that the code did or did not meet the acceptance criteria as stated in the code's VVP.

Codes under test:
Provide a brief description of the code's purpose. Include executable information and identify the location in which the regression test inputs and outputs are archived.

Test Methodology:
Describe how the code will be compared to the previous results.

Identify and describe the comparison utility used to compare the results of the regression test results to previous results. Describe the format of the comparison output file, and how differences are to be evaluated. Provide enough detail so that the reader can evaluate the comparison output files.

If a manual comparison is to be performed, describe how that will be accomplished and how differences will be documented and evaluated.

Test Results:
The test results section should provide the reader with the information necessary to evaluate each test case and determine if the test case performs as expected. The following should be included in the test results section:

- A listing of the test case(s) input and output files names.
- A description of the test case(s) acceptance criteria.
- A listing of the test case(s) comparison results should be part of the report, so the reader can evaluate the results.
- A discussion of the regression testing results, including a statement of acceptability.

References:
Provide references for all documents referred to within the report.

This work of authorship was prepared as an account of work sponsored by an agency of the United States Government. Accordingly, the United States Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so for United States Government purposes. Neither Sandia Corporation, the United States Government, nor any agency thereof, nor any of their employees makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by Sandia Corporation, the United States Government, or any agency thereof. The views and opinions expressed herein do not necessarily state or reflect those of Sandia Corporation, the United States Government or any agency thereof.

Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Parties are allowed to download copies at no cost for internal use within your organization only provided that any copies made are true and accurate. Copies must include a statement acknowledging Sandia Corporation's authorship of the subject matter.