

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 12

Effective Date: 06/21/06

Author: Mario Chavez Mario Chavez 6/21/06
(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 194, 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 6.2 A.) 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.

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.

Note: Specific utility & library applications written for use within these types of software that can be verified by hand calculations or manual inspection shall be covered by and meet the requirements of NP 9-1.

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 - The Lead Code Sponsor, Code Subject Matter Expert, and the Code Developer make up the team, which can expand or shrink as necessary depending on the complexity of the development effort. Individual(s) who oversees the Software Quality Assurance (SQA) process for a particular software item.

Code Subject Matter Expert - This individual maintains responsibility for the technical quality of codes utilized for a specific subject matter area.

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.

Contracted Software - Individuals or organizations developing and supplying software under contract.

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, data base 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.

Lead Code Sponsor - This individual(s) is responsible for coordinating the software development process and providing software development technology of all codes supporting the PA project.

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.

Life-Cycle Review - The process of assessing the baseline documentation to verify that the products of a software development phase meet the requirements defined for it by previous phases.

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.

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

Patch Change - A change that is of low complexity and easy to verify its implementation (e.g., a change to re-compile and link to a modified include file without changing the source, a change to a database view to expand a field from 8 to 10 characters, etc.).

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. Assumed to have read the appropriate QA documentation and analysis or test plan as applicable.

Regression Testing - Software testing conducted during installation and checkout or after there has been a significant system software or hardware change to verify that the software produces the same results for a given set of inputs as previously documented.

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

Software Review Board (SRB) - The Technical Reviewer, the Responsible Manager, the Code Team/Sponsor, and the SCM Coordinator make up the SRB, which can expand as necessary to cover the workload and supply needed expertise. The SRB approves Software Quality Assurance Plans (SQAPs), provides change control prior to procurement and/or development of software, pre-approves

the use and implementation of a peer review process for software validation, and reviews and approves software QA documentation.

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 some 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 Reviewer - A team member responsible for the evaluation of the V&V activities for each phase of the software lifecycle.

User - A person who uses baseline software.

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 at the end of each software development 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 three parts as needed, each separated by a period. This system is described below:

Version **X.Y.Z** X is the major field. Y and Z are the minor field(s), where Z is used for patches. X, Y, and Z can be alpha-numeric characters of any length, e.g., 2.3.8, 1.01.C, 12.2B. 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.

Note: While software qualified under a previous version of NP 19-1 is undergoing change, it needs to be evaluated on a case-by-case basis for appropriate classification, i.e., either acquired or developed and version identifier assignment as appropriate.

2.2 Software Qualification

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

Table 1. Compliance Decision (CD) Software Requirements

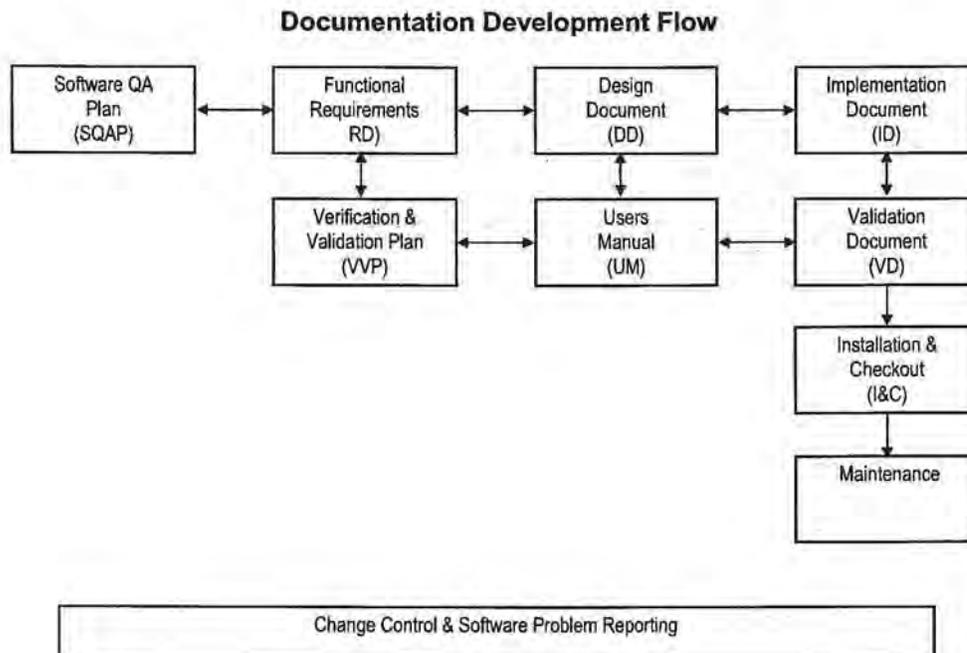
PHASE	Planning	Requirements		Design	Implementation		Validation	I&C			Maintenance		Retirement
LOCATION	2.3.1	2.3.2		2.3.3	2.3.4		2.3.5	2.3.6			2.3.7 ⁵		2.3.8
APPLICABLE DOCUMENT	SQAP	RD ³	VVP	DD ⁴	ID	UM ²	VD	I & C	A C	A U	CC ^a	SPR	
FORMS ¹ NP 19-1-X	1	2	3	4	5	6	7	8	-	-	9	10	-
Acquired	X	X	X	-	^b	X	X	X	X	X	X	X	X
Developed	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 and SPR 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 significant, 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 significant in nature, then a detailed design document is required only for the new portions of the design.
5. Change Control (CC) and Software Problem Reporting (SPR) are required as needed (i.e., when changes to baseline documents are needed or when bugs are discovered.)

Figure 1**2.2.1 Acquired Software**

Commercial Off the Shelf Software (Excel, Mathematica, Word etc.) as received is acquired software and is exempt from the software development requirements. Numerical Modeling software may be acquired software but is not exempt from the software development requirements. The acquisition of Contracted Software shall follow NP 4-1 "Procurement".

Prior to use, Acquired Software shall be evaluated against the life cycle phases. The process of qualifying such software for use is provided below.

The Code Team/Sponsor shall review the adequacy of the primitive baseline to determine the following in accordance with the QA measures identified in Table 1 for Acquired Software:

- A. Adequacy of existing verification and validation and software documentation to support operation and maintenance is defined by criteria found in Sections 2.3.1 through 2.3.8.
- B. Activities to be performed and the documentation necessary to accept the software for its intended use and place it under configuration control is defined by criteria found in Sections 2.3.1 through 2.3.8.

The following documents are produced following the process described in Appendix A:

- The Primitive Baseline consists of the Software QA Plan (see section 2.3.1) and existing software documentation.
- The Software QA Plan is written by the Code Team/Sponsor and will include a comparison of the existing documentation with the software life cycle requirements of this procedure.
- The Code Team/Sponsor shall develop and submit the Software QA plan in accordance with section 2.3.1.

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 under configuration control and developed within the scope of these QA requirements will not require a stand alone SQAP.

The SQAP shall identify:

- The software to which it applies, objectives of the software, problem statements, 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 existing documentation with the software life cycle requirements of this procedure.
- If any deviations from the documentation required by NP 19-1 are anticipated, e.g., a database may not use an Implementation 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.

SQAP may be written for an individual code or a set of codes. It should be developed by Code Team/Sponsor and approved (by signature) by the Responsible Manager, Technical Reviewer(s), 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 following the process described in Appendix A, 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).

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 following the process described in Appendix A, using the phase criteria listed on the Design Document Criteria Form NP 19-1-4, (Appendix E). 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 and the process of generating executable software, and
- The User's Manual (UM) - provides information to assist users 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 following the process described in Appendix A 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).

2.3.5 Validation Phase

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 following the process described in Appendix A, using the phase criteria listed on the VD Document Criteria Form NP 19-1-7, (Appendix H).

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 provides evidence of:

- The execution of the validation cases on the production computer,
- The installation of the baseline software on the production computer (re-compiling and linking if necessary), and
- The performance of 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 Code Team/Sponsor shall produce the I&C Form, NP 19-1-8 (Appendix I) following the process described in Appendix A.

Note: When Programmatic Decision (PD) Software (Section 2.4) is installed, an Installation and Checkout Form and the Implementation Document is submitted to the Software Configuration Management Coordinator by the Code Team/Sponsor. Since completion of required software QA

documentation may require critical code modifications, the results of analyses using PD software may not be used for WIPP compliance decisions.

Note: Installation on a network of identical computers running identical operating systems requires testing on only one of the machines.

Note: If testing for the Validation Phase in Section 2.3.4 was performed on the production computer, then the test cases need not be rerun (provide reference to the VD).

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 following the process in Appendix A. When specifying access control on a system-wide basis, document or provide reference to the Access Control Memorandum describing system specific controls.

2.3.6.3 Approved Users Memorandum

The Approved Users Memorandum identifies users for a particular code. Users may be identified by name, organization, group, readers of approved test and/or analysis plan, etc. The Approved Users Memorandum shall be included as part of Installation and Checkout Phase Documentation.

The Code Team/Sponsor shall document approved users in the Approved Users Memorandum following the process in Appendix A.

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, following the process in Appendix A and using the Change Control Form, Form NP 19-1-9, (Appendix J),

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.

Patch changes – can be used for very small fixes to the code usually one or two lines of source code or expanding a fields character length etc. Patch changes can be documented and tested with the Change Control Form and Installation & Checkout Form.

The SCM Coordinator shall:

- identify affected software configuration baselines.
- verify unique revision identifier.
- inform affected users of approved changes. **Note:** If an organization is listed as an approved user, the organization's manager will be notified.
- redline/update baseline list.
- maintain a copy of the Change Control Form and forward to the SNL WIPP Records Center.

The Code Team/Sponsor shall:

- Perform modifications to software and/or associated baseline documentation 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 component identifiers are consistent (see Section 2.0)
- If modifications require re-compilation of the software, perform regression testing as identified in the approved VVP. 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.

Note: If the software was modified to correct a problem, Code Team/Sponsor shall ensure that the Software Problem Reporting (SPR) process (Section 2.3.7.3) has been initiated.

2.3.7.2 System Software and Hardware Change Control

2.3.7.2.1 Coding Documentation Standards

Any change to software must be accompanied by documentation describing the change, the date the change was made, and the name of the person responsible for implementing the change. This documentation should be clearly identified, and placed in the code in the vicinity of the change, as well as at the top of the code prior to the first executable line. The code reviewer shall determine if this documentation is clear and sufficient.

2.3.7.2.2 Significant System Software or Hardware Changes

The Code Team/Sponsor (single-user systems) or System Administrator (multi-user systems) shall propose significant system software or hardware changes following the process described in Appendix A, using the Change Control Form NP 19-1-9 (Appendix J).

Examples of significant 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

In general, changes are significant if they impact the results generated by production software or cause recompilation of production software.

The Code Team/Sponsor or System Administrator shall:

- perform the approved system modification to the system software and/or hardware.

- perform regression testing (after significant 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.

2.3.7.3 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).

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 the software or analysis, which used the output, produced by the subject software version. If additional calculations are needed or the analysis is to be redone, follow NP 9-1 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, initiate a CAR per NP 16-1. The evaluation and resolution of the software problem shall be documented on field 4 of the Software Problem Report Form (attach pages as needed).

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

The responsible manager shall approve the evaluation and resolution by signing the form and forwarding it to the SCM Coordinator.

The SCM Coordinator shall assign an SPR number of the Software Problem Report. The SCM Coordinator shall also update/redline the Software Baseline List. The SCM Coordinator shall retain copies of the form, and forward a copy 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.

2.3.7.4 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 process for preparation and approval of software baselines is described in Appendix A.

The SCM Coordinator shall maintain a Software Baseline List, and make it available upon request. The SCM Coordinator performs a completeness review to ensure compliance with the procedure, and 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 (see Appendix A),
- RD version,
- VVP version,
- DD version,
- ID version,
- UM version,
- VD version,
- list of approved users (may be listed by name, organization, group, or task, etc...)
- list of approved system software/hardware configurations,
- list of outstanding 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 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.

2.4 Interim use of Unqualified Software to Support Programmatic Decisions

With written permission granted in advance by the Sandia Carlsbad Programs Group Manager relying on input from the Responsible Manager and Software 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 a 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 Carlsbad Programs Group Manager or designee:

Approve the Analysis Plan by signing the AP. **NOTE:** This signature serves as the written permission.

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 QA 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

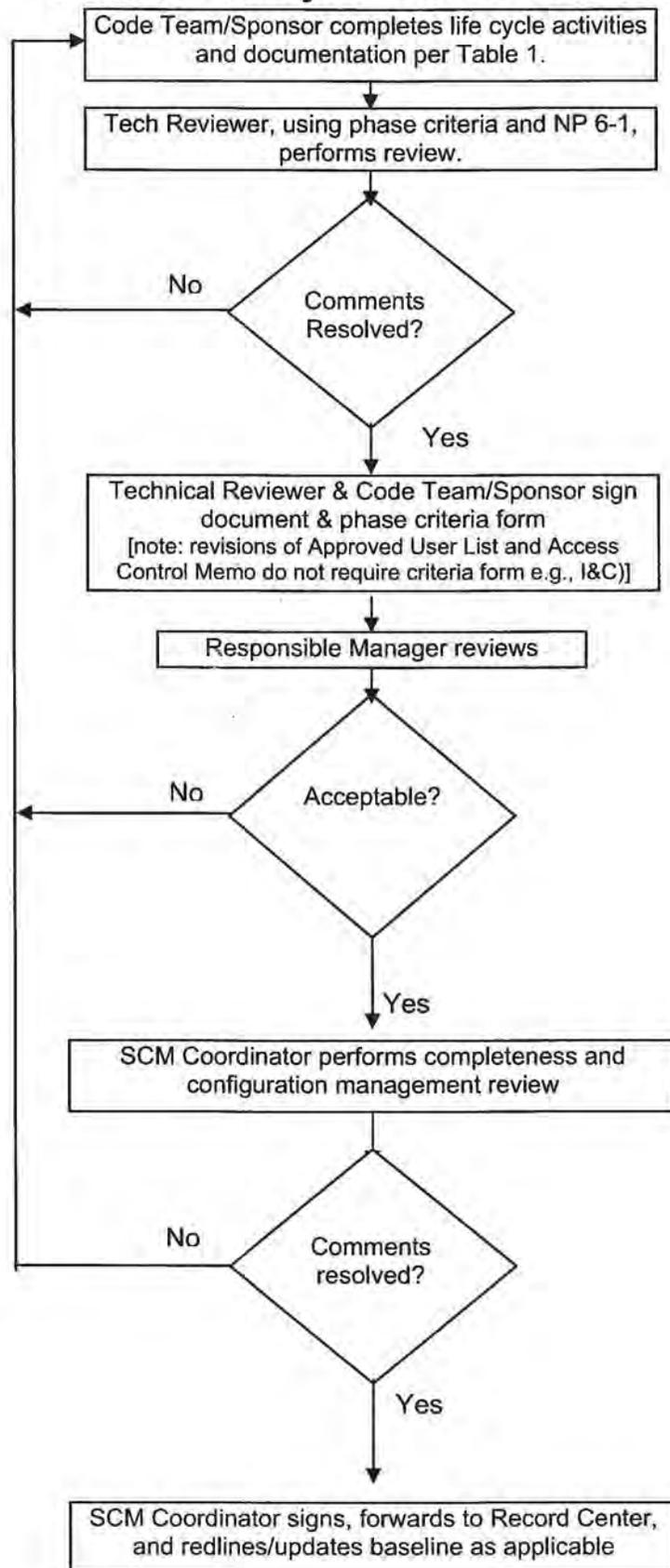
QA Record

- 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)
- Requirement 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)
- Access Control Memorandum
- Approved User Change Memorandum
- Code Retirement Request Memorandum

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 A Software Life-Cycle Process Flow Chart



Appendix B

<p style="text-align: center;">NUCLEAR WASTE MANAGEMENT PROCEDURE</p> <p>Sandia National Laboratories</p>	<h2 style="margin: 0;">Software QA Plan Criteria</h2>	<p>Form Number: NP 19-1-1</p> <p>Page 1 of 1</p>
--	---	--

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.

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

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

Key for check boxes above:

Check **Yes** for each item reviewed and found acceptable.

Check **N/A** for items which are not applicable.

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 includes a quantification of acceptable error range per %. Acceptance criteria specify the outputs and features required to demonstrate acceptable performance and provide a comparative basis for each required output or feature to be validated
 - **Maintainability.** The structure and style of the requirements allow the necessary changes can be made.
9. **External Interfaces.** Describe any interaction 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.** All 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.

Appendix E

<p style="text-align: center;">NUCLEAR WASTE MANAGEMENT PROCEDURE</p> <p><small>Sandia National Laboratories</small></p>	<h2 style="margin: 0;">Design Document Criteria</h2>	<p>Form Number: NP 19-1-4</p> <p>Page 1 of 1</p>
---	--	--

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. Allowable or Prescribed Ranges for Inputs and Outputs Yes
8. Verifiability: Is the design verifiable through testing or other means? Yes
9. Consistency and Traceability: Is the design consistent with and traceable to the software's requirements? Yes
10. Technical Feasibility: Is the design technically feasible? Yes
11. Implementability: Is the design presented in sufficient detail to allow implementation as computer software? Yes

12. Code Team/Sponsor <i>(print)</i> _____	_____ Signature	_____ Date
13. Technical Reviewer <i>(print)</i> _____	_____ Signature	_____ Date
14. Responsible Manager <i>(print)</i> _____	_____ Signature	_____ Date
15. SCM Coordinator <i>(print)</i> _____	_____ Signature	_____ Date

Key for check boxes above:

Check **Yes** for each item reviewed and found acceptable

Appendix G

<p style="text-align: center;">NUCLEAR WASTE MANAGEMENT PROCEDURE</p> <p>Sandia National Laboratories</p>	<h2 style="margin: 0;">User's Manual Criteria</h2>	<p>Form Number: NP 19-1-6</p> <p>Page 1 of 1</p>
--	--	--

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 sponsor 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 N/A
6. An explanation of the mathematical model and numerical models, where applicable as based on code functionality? Yes N/A
7. Physical and mathematical assumptions, where applicable based on code functionality? Yes N/A
8. The capabilities and limitations inherent in the software? Yes
9. Instructions that describe the user's interaction with the software? Yes
10. The identification of input parameters, formats, and valid ranges? Yes
11. Messages initiated as a result of improper input and how the user can respond? Yes
12. The identification and description of output specifications and formats? Yes
13. A description of any required training necessary to use the software? Yes
14. The identification of components of the code that were not tested? Yes

15.	Code Team/Sponsor <i>(print)</i>	Signature	Date
16.	Technical Reviewer <i>(print)</i>	Signature	Date
17.	Responsible Manager <i>(print)</i>	Signature	Date
18.	SCM Coordinator <i>(print)</i>	Signature	Date

Key for check boxes above:

Check Yes for each item reviewed and found acceptable
Check N/A for items not applicable, where applicable as based on code functionality

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 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 usage as defined by the User's Manual.

Sample

Appendix I

NUCLEAR WASTE MANAGEMENT PROCEDURE Sandia National Laboratories	<h2 style="margin: 0;">Software Installation and Checkout</h2>	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 Revision identifier ERMS#: _____

b. VD Document/Revision identifier 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 SPR outstanding: Yes No SPR No(s): _____

Test Case Information

13. Directory/Path to ERMS library: _____

14. Schedules: _____

15. Libraries: _____

16. Input Files: _____

17. Output Files: _____

18. Test Evaluation:
 Test results fully met specified acceptance criteria Yes

19. Access Control and Approved User Memo are attached to the I&C or are referenced:
 attached, referenced, ERMS# _____

20.	Code Team/Sponsor (print)	Signature	Date
21.	Technical Reviewer (print)	Signature	Date
22.	Responsible Manager (print)	Signature	Date
23.	SCM Coordinator (print)	Signature	Date

Appendix J

<p style="text-align: center;">NUCLEAR WASTE MANAGEMENT PROCEDURE</p> <p style="font-size: small;">Sandia National Laboratories</p>	<h2 style="margin: 0;">Change Control (Software/Hardware/Baseline Document)</h2>	<p>Form Number: NP 19-1-9</p> <p>Page ____ of ____</p>
--	--	---

1. Software Name: _____
2. Software Version Identifier a) Current: _____ b) Proposed: _____
3. Software Classification a) Current: _____ b) Proposed: _____
4. ERMS # _____
5. Hardware/Software Platform: _____
6. Type of change: Major Minor Patch
7. SPR No.(s) (if applicable): _____
8. Proposed Changes: (attach pages as needed or use continuation sheet of form)

Sample Form

Software Plan (QA) Document Affected Required Resolution

Yes No* Revision Page Change Addenda

Version No: _____
New Version No: _____
*Rationale _____

Requirements Document (RD) Yes No* Revision Page Change Addenda

Version No: _____
New Version No: _____
*Rationale _____

Verification and Validation Plan (VVP) Yes No* Revision Page Change Addenda

Version No: _____
New Version No: _____
*Rationale _____

Design Document (DD) Yes No* Revision Page Change Addenda

Version No: _____
New Version No: _____
*Rationale _____

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: _____

9. System Software/Hardware Change Section:

10. _____
Code Team/Sponsor's Name (print) or Computer Administrator
Signature Date

11. _____
Technical Reviewer's Name (print)
Signature Date

12. _____
Responsible Manager's Name (print)
Signature Date

13. _____
SCM Coordinator's Name (print)
Signature Date

Appendix J (continued)

Change Control (Software/Hardware/Baseline Document) (continuation sheet)

Form Number:
NP 19-1-9

Page ____ of ____

Continuation of Item 8, 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. For proposed system changes, enter applicable information (e.g. operating system name, hardware, device, etc.).
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. For proposed system changes, enter current status information (e.g. operating system version).
3. **Software Classification:** On (a) enter the current classification (i.e. acquired, developed) and on (b) enter the proposed classification.
4. **ERMS # assigned to Change Control Form,** assigned by SMM Coordinator.
5. **Hardware/Software Platform:** Enter the hardware platform on which the software 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 the baseline to indicate whether changes are major, minor, or patch.
7. **Provide the software problem report (SPR) numbers** if this change is to address any SPR(s).
8. **Proposed Changes:** Use this section to describe in detail the changes each document will be updated. 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.

Implementation Document (ID): Check if this document is affected and how it will be updated. In general, all ID changes will be revisions, not addenda's. Describe what aspects of the coding will change.

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 rationale 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.

User Manuals (UM): Describe what user instructions will be changed, added, or deleted.

9. 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.

10. Code Sponsor or System Administrator Signature.

Code Sponsor signs for changes to baseline software.

System Administrator signs for changes to system software/hardware.

11. Technical Reviewer Signature. Indicated concurrence with impact to baseline software/hardware. For system software / hardware indicates concurrence with evaluation of impact to production baseline codes.

12. Responsible Manager signature. After signing form, RM forwards to SCM Coordinator.

13. SCM Coordinator Signature. SCM Coordinator signs change control form or returns it to code sponsor or computer administrator for proper completion. After SCM coordinator signature, forwards form to approved user and RM.

Sample Form

Appendix K

<p style="text-align: center;">NUCLEAR WASTE MANAGEMENT PROCEDURE</p> <p><small>Sandia National Laboratories</small></p>	<h2 style="margin: 0;">Software Problem Report (SPR)</h2>	<p>Form Number: NP 19-1-10</p> <p>Page ____ of ____</p>															
<p>1. Software Name and Version: _____</p>																	
<p>2. SPR Classification:</p> <p><input type="checkbox"/> 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)</p> <p><input type="checkbox"/> Minor (Everything else)</p>																	
<p>3. Summary of Error: [how to reproduce it and suggestions for fixing it (optional)] <small>(each page is a sheet or use continuation sheet at end of form)</small></p>																	
<p>4. <small>Support Analysis (report reference with ERMS# and a decision to re-do or use the analysis as is) (attach pages as needed use continuation sheet at end of form)</small></p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: left;"><u>Title</u></th> <th style="width: 20%; text-align: center;"><u>ERMS#</u></th> <th style="width: 30%; text-align: center;"><u>Decision</u></th> </tr> </thead> <tbody> <tr><td>_____</td><td>_____</td><td>_____</td></tr> <tr><td>_____</td><td>_____</td><td>_____</td></tr> <tr><td>_____</td><td>_____</td><td>_____</td></tr> <tr><td>_____</td><td>_____</td><td>_____</td></tr> </tbody> </table>			<u>Title</u>	<u>ERMS#</u>	<u>Decision</u>	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____
<u>Title</u>	<u>ERMS#</u>	<u>Decision</u>															
_____	_____	_____															
_____	_____	_____															
_____	_____	_____															
_____	_____	_____															
<p>5. Code Team/Sponsor Name (<i>print</i>) _____ Signature _____ Date _____</p>																	
<p>6. Technical Reviewer Name (<i>print</i>) _____ Signature _____ Date _____</p>																	
<p>7. Responsible Manager Name (<i>print</i>) _____ Signature _____ Date _____</p>																	
<p>8. SCM Coordinator Name (<i>print</i>) _____ Signature _____ Date _____</p>																	
<p>9. SPR No. (Year and sequence, e.g., 2004-01): _____</p>																	

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: <p style="text-align: center; font-size: 48px; opacity: 0.5; transform: rotate(-15deg);">Sample Form</p>	

NOTICE: This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness or 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 the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors.

This document was authored by Sandia Corporation under Contract No. DE-AC04-94AL85000 with the United States Department of Energy's National Nuclear Security Administration. 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.