Approval: Shannon A. Brennan Date: 03-13-08
Manager, National Spent Nuclear Fuel Program

I. PURPOSE AND SCOPE

Software used by the NSNFP to generate or manipulate National Spent Nuclear Fuel Program (NSNFP) technical information is controlled to ensure that calculations are properly performed and that the information integrity is not compromised.

This procedure applies to the purchase, development, modification, and use of software used to generate technical analysis results by the NSFNP. This procedure does not apply to word processors, spreadsheets, database managers, e-mail, and other types of automated office support systems unless modified by the NSFNP. However, applications (i.e., software routines and macros) generated within these types of software are subject to limited requirements as provided in this procedure.

Confidential controls permitting authorized access and preventing unauthorized access to computer systems are implemented by NSNFP.

The NSNFP Spent Fuel Database is controlled in accordance with procedure NSNFP 19.02.

II. SUMMARY

This procedure governs software use within NSNFP. For software controlled by the Yucca Mountain Project or for software routines and macros, this procedure ensures that software is installed, checked, and used in accordance with the appropriate governing documents. For other software controlled by NSFNP procedures, it directs the development of a software plan that specifies the requirements, controls, tests, and user information for each software product. The software plan is placed under configuration management and becomes the baseline for its respective software product.

III. PROCEDURE

A. Software Routines and Macros within Exempt Commercial Software

PSO Technical Staff

1. Create or modify routines and macros as needed.

2. Ensure in coordination with the Document Control Coordinator (DCC) that software routines and macros have a unique identifier, which includes the version of the software application.

3. Independently verify by inspection or alternate calculation that the software routines and macros calculate correct results. Tests and test results from reviews and verifications shall be included in the *acceptance testing* (see glossary) documentation. Tests conducted as reviews or verifications do not substitute for performing comprehensive, end-of-development acceptance tests.

PSO Technical Staff

4. Within documentation of technical products generated using the software (see NSNFP Procedure 3.04, Engineering Documentation), reference the software and the associated routines and macros along with documentation of verification performed in previous step of this procedure.

5. GO TO Step III.D.

B. Software Controlled by Yucca Mountain Project

PSO Technical Staff

1. If software will be used without modification, GO TO Step III.D.

2. Otherwise, request changes in accordance with governing Yucca Mountain Project procedures.

3. After necessary changes have been completed, GO TO Step III.D.

C. Other Software

PSO Technical Staff

1. If software has an active software plan and will be used without modification, GO TO Step III.D.

2. Establish unique identifier in coordination with the DCC.

3. For software developed outside of the NSNFP, perform the following:

   a. Prepare software plan in accordance with Attachment A, indicating how each criterion is satisfied and providing justification for any deemed not applicable.

   b. Review and issue software plan in accordance with NSNFP Procedure 6.01.

   c. Acquire software that satisfies the software plan, using NSFNP Procedure 4.02 for any procurement. Tests and test results from reviews and verifications shall be included in the acceptance test documentation. Tests conducted as reviews or verifications do not substitute for performing comprehensive, end-of-development acceptance tests.

   d. Go to III.D.

4. To revise or develop software within NSNFP, prepare or revise and execute a software plan.

   a. Perform planning and requirements phase of software development.

      (1) Prepare or revise planning and requirements phase of software development and document as shown in Section 1 of Attachment A.

PSO Technical Staff

(2) Verify satisfactory completion of the design phase by reviewing and issuing the document in accordance with NSFNP Procedure 6.01.

b. Perform design phase of software development.

(1) As outlined in Section 2 of Attachment A, document the new or revised design specifications to ensure specified requirements are satisfied.

(2) Verify satisfactory completion of the design phase by reviewing and issuing the document in accordance with NSFNP Procedure 6.01.

c. Perform implementation phase of software development.

(1) Create or modify code in accordance with software plan.

(2) Establish unique identifier for the software in coordination with the DCC.

(3) Prepare user information as instructed in Section 3 of Attachment A.

(4) Verify satisfactory completion of the implementation phase by reviewing and issuing document in accordance with NSFNP Procedure 6.01.

d. Perform testing phase of software development.

(1) Specify tests required for *software validation* (see glossary) and installation as instructed in Section 4 of Attachment A.

(2) Review and issue test plans in accordance with NSFNP Procedure 6.01.

(3) Perform validation testing in accordance with approved test plans.

(4) If validation tests were unsuccessful, perform the following:

(a) Document software defect and its resolution in test documentation.

(b) Repeat applicable steps to revise the software or software plan as needed.

PSO Technical Staff

(5) Submit documentation of successful validation to the DCC along with test documentation generated as part of validation tests and final (released) version of software. Tests and test results from reviews and verifications shall be included in the acceptance test documentation. Tests conducted as reviews or verifications do not substitute for performing comprehensive, end-of-development acceptance tests.

D. Software Installation, Checkout, and Use

PSO Technical Staff

1. As applicable, obtain controlled copy of software and related documents from the DCC. *Support software* (see glossary) is not qualified nor is it baselined. However, such software shall be placed under configuration management control (including change control). At a minimum, configuration items to be controlled shall include documentation, computer programs and support software.

2. In coordination with the DCC, ensure that the software and related documents are uniquely identified.

3. Independently review the selected software to ensure it is suitable for the problem being solved. If the intended use of the software item will require the use of inputs outside the ranges verified during validation testing, the appropriate baseline elements shall be reverified and revalidated for the expected range of inputs prior to continuing use.

4. Ensure software installation and installation tests are performed as specified in its user documentation or software plan.

5. If software cannot be successfully installed, prepare and disposition a Software Defect Notification, Form 19.01-1, according to the instructions on the form.

6. If software has not been validated for the current application or if software will be used outside its range of validation as baselined, perform additional validation testing in accordance with its software plan or Step III.C.3.d.

7. Submit software test documentation generated as part of installation tests, to the DCC.

8. Unless software is controlled by the Yucca Mountain Project or is the same version as that currently controlled by the DCC, submit copy of software to the DCC.

9. Use software in accordance with its user documentation.

10. If software deficiencies are identified, the individual identifying the problem, in coordination with the software developer,prepare and disposition a Software Defect Notification (Form 19.01-1) in accordance with the instructions on the

form determines if the problem is an error or other type of problem (e.g., user mistake) by

a. Verifying that input data were entered correctly,

b. Considering other potential user mistakes to verify that the deficiency is inherently attributable to the software, and

c. Reproducing the deficiency under controlled conditions as appropriate to assess the scope of the problem, establish the impact on previous work, eliminate any computer-specific conflicts, and identify an acceptable resolution (including any interim measures to be taken).

11. If a deficiency is verified to be a software error, the individual identifying the problem prepares a Software Defect Report (Form 19.01-1) in accordance with the instructions on the form.

12. DCC notifies affected users or organizations and includes the pertinent information in the software listing.

13. The Technical Lead dispositions the problem report and notifies the originator of the results of corrective action closure.

14. Software requirements shall be traceable throughout the *software life cycle* (see glossary). *Traceability* (see glossary) shall be documented..

E. Software Retirement

Technical Lead

1. When the software will no longer be used, inactivate its software plan by canceling it in accordance with NSFNP Procedure 6.01.

IV. REFERENCES

None.

V. DEFINITIONS

Terms appearing in italics followed by the notation "see glossary" are defined in the NSNFP Documents Manual Introduction and Glossary.

VI. ATTACHMENTS

Attachment A, Development and Review Criteria for Software Plan

VII. QUALITY RECORDS

The following quality records generated as a result of this procedure require retention according to the identified classification and NSNFP Procedure 17.01.

Lifetime

A. Software Plan and implementing documents

B. Software Defect Notification

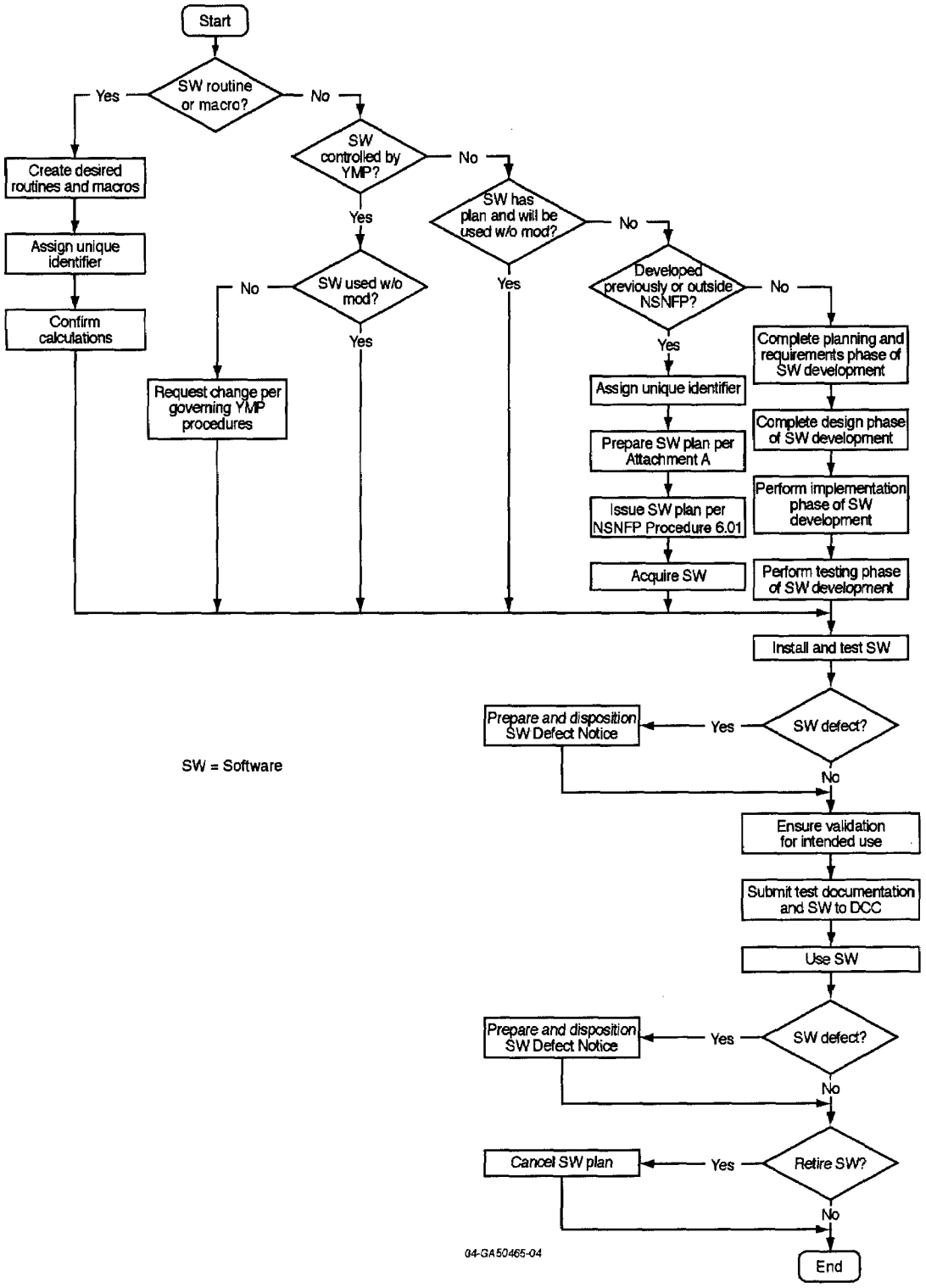
C. Test documentation

D. Software

Nonpermanent

None.

VIII. PROCEDURE FLOW DIAGRAM



Attachment A

Development and Review Criteria for Software Plan

The Development and Review Criteria are presented in the suggested format for a software plan prepared as a single document. The software plan may also be composed of individual documents that address each of the development and review criteria. Existing documentation may be incorporated into the software plan by reference. Documents referenced within the software plan, when such reference is for the purpose of satisfying the criteria, must be controlled by a QARD-compliant document control system and must be reviewed to assure that software plan criteria are satisfied. The minimum reviewers for each section are shown in parentheses following the section header.

Verification following requirements, design, implementation, and testing is documented in the review performed following each of the software development phases.

1. Planning and Requirements

Purpose, Scope, and Applicability (Qualified[a] Software Reviewer)

Identify the specific software governed by the plan and its overall nature and purpose are described along with the basis for applying software controls. Software objectives, scope, and applicability are appropriate for the activity, and the applicable Work Breakdown Structure number is referenced.

Quality Assurance (NSNFP Quality Engineer)

The following text is included:

> Quality assurance is maintained by compliance with NSFNP Procedure 19.01, Software Control. NSFNP Procedure 19.01 prescribes processes and associated organizational responsibilities for development, verification, validation, and acceptance along with the associated responsibilities for achieving software quality. NSFNP Procedure 19.01 directs software design requirements, design specifications (including testing requirements), documentation of verifications following each of the software development phases, and validation at the completion of product development to be baselined in the Software Plan, which, upon completion, becomes the software baseline document. The software plan is controlled in accordance with NSFNP Procedure 6.01.

The following are clearly identified:

- The organizations responsible for development, verification, validation, and acceptance along with their respective responsibilities for achieving software quality
- Applicable standards, conventions, methodologies or other requirements

---

a. Qualified in accordance with applicable NSNFP procedures and not associated with development of the software.

Attachment A

- Any required reviews, holdpoints, and associated criteria.

Requirements (Qualified Software Reviewer)

Requirements, which are specified within the subsections below, satisfy specified QA requirements and provide sufficient detail to enable successful design and/or acquisition of the software to achieve the planned objectives. Achievement of the specified requirements can be verified and validated.

Functional Requirements

The functions that the software must perform and any design constraints are clearly specified.

Performance Requirements and Other Attributes

Requirements, if any, for the time-related issues of software operation such as speed, recovery time, and response time are clearly specified along with other essential attributes such as portability, access control, and maintainability. Measures to mitigate the consequences of potential problems shall be an integral part of the design. These potential problems include external and internal abnormal conditions and events that can affect the computer program. including the software operating environment for office computer use.

Interfaces

Requirements to ensure successful interface with users and other hardware or software interfaces are clearly specified. The software design shall consider the computer *software operating environment* (see glossary).

Control Points

The following sentence is included:

> Software is documented, reviewed, and baselined in accordance with NSFNP Procedure 19.01, Software Control.

If necessary to ensure safety, holdpoints (in addition to those stipulated by NSFNP Procedure 19.01) are specified.

Attachment A

2. **Design** (Qualified[a] Software Reviewer)

Design Specifications

The following are clearly specified:

- The major components of the software design as they relate to the design requirements.
- The theoretical basis, mathematical model, control flow, data flow, control logic, and data structure employed by the software.
- The allowed range of inputs and output.
- The design—in a manner that can be translated into code.
- Design-based test cases to demonstrate the software produces the correct results within the specified range of inputs. These tests may use hand calculations, other validated software, experiments and tests, problems with known solutions, and comparisons with confirmed published data and correlation.

Design is specified in a manner that allows traceability to requirements and is sufficient to ensure adherence to the requirements.

Design Documentation

The design documentation shall specify the essential features of the software, including the following as appropriate:

a. A description of the major components of the software design as they relate to the software requirements.

b. A technical description of the software with respect to the theoretical basis, mathematical model, control flow, data flow, control logic, and data structure.

c. A description of the allowable or defined ranges for inputs and outputs.

d. The design, described in a manner that can be translated into code.

e. The generation of design-based test cases.

f. The generation of test plans/cases, based on the requirements and design, shall provide for acceptance criteria and verification of results.

g. For those computer programs used in design activities, the test plans shall provide for ensuring that the software produces correct results. For those computer programs used for

Attachment A

operational control, computer test plans shall provide for demonstrating required performance over the range of operation of the controlled function or process.

3. **Implementation** (Qualified[a] Software Reviewer)

Implementation satisfies requirements and design specifications.

User information contains an overview of user interaction with the software and specifies any background, training, or other prerequisites that are helpful or required for use of the software. In addition, this section provides:

- The form, content, and associated limitations (i.e., allowable ranges) for user inputs and software outputs
- Default values for user inputs and file formats for any input and output data files
- Sample problems along with anticipated errors and appropriate user responses
- Any hardware and software requirements, installation instructions, and installation tests

4. **Testing** (Qualified[a] Software Reviewer)

Tests confirm implementation satisfies software design requirements.

Test cases and associated acceptance criteria (based on the requirements and the design) are provided to verify that code and user documentation meet the design and implementation objectives. Test plans will:

- Identify allowable hardware and software configurations
- Specify the recording of results of the validation
- Include results of reviews and tests
- Require regression testing for software modifications to detect unintended adverse effects introduced during the modification of the software, to verify that the modifications have not caused unintended adverse effects, and to verify that a modified software still meets specified requirements.
- Specify minimum tests to be performed for each software installation, modification, or change to the operating system
- Specify documentation (Technical Report or Engineering Design File according to NSNFP Procedure 3.04) needed to document successful completion of validation and installation tests.

Attachment A

Tests confirm implementation satisfies software design requirements.

- Tests and test documentation are organized in a manner that allows traceability to requirements and is sufficient to ensure adherence to the requirements.

- Software verification shall include review of the test results. Software verification shall be completed prior to approval of the computer program for use.

- Testing shall be performed to an approved plan or process and on different computers with all operating environments with which the software may be used to ensure adherence to the requirements and to ensure the software produces correct results for test cases.

- Testing shall demonstrate, as appropriate, that the computer program:

  a. Properly handles abnormal conditions and events as well as failures.

  b. Does not perform adverse unintended functions. Observations of unexpected or unintended results shall be documented and dispositioned prior to test result approval.

  c. Does not unexpectedly degrade the system either by itself or in combination with other functions or configuration items.