1. PURPOSE

When prompted by MCP-2374, "Analysis and Calculations," this procedure is performed to ensure that *analysis software* (see def.) adequately and correctly performs all intended functions, thereby providing accurate analysis results.

2. SCOPE

This procedure contains the instructions for determining software needs, controlling and documenting validation activities, reporting errors, retiring analysis software, and maintaining and updating the software library.

This procedure applies to analysis software used to perform calculations.

This procedure does not apply to:

- Analysis software that can be independently verified by visual inspection or hand calculation during the checking process per the directions in MCP-2374
- Analysis software used to perform calculations that are part of a project with a non-INEEL participant (such as work for others) where requirements are specified and documented by the customer
- Analysis software that is being developed or modified; MCP-550 is used to develop or modify analysis software, routines, and macros.

Appendix A outlines the analysis software control process.

3. RESPONSIBILITIES

3.1 Responsibilities

Performer	Responsibilities
Functional Manager/ Project Engineer, Program Manager	Identify the need to control the software in accordance with this procedure.
Analysis Performer	Assume overall responsibility for ensuring analysis software, used to perform analysis activities, is identified, validated, and controlled in accordance with this procedure prior to use.

Application Trustee	Perform necessary activities to put analysis software and validation documents and records under configuration control. Assume technical responsibility for the software. Identify, document, and see that software errors are resolved. Retire software. Be the primary point of contact and act as an information resource on the technical attributes of the software. Ensure software information is included in the *Software Inventory Database* (see def). Be knowledgeable of software licensing agreements. Review and approve validation plans. Ensure software problems and errors are resolved. Periodically review the software supplier's homepage and install supplier patches and plug-ins, when appropriate. Retire software, when appropriate.
Validation Tester	Perform and document software validation activities.
Software Librarian	Support the Application Trustee by entering software data and validation documentation into the Software Inventory Database.

4. INSTRUCTIONS

4.1 Determining Software Needs

4.1.1 Analysis Performer: Review the *Software Inventory Database* (http://juneau:81/earch/start.html) to determine if the software is currently available and configuration controlled under this management control procedure.

4.1.2 Begin the software control process by completing Section I – Software Functional Description, of Form 562.19, "Software Lifecycle Plan and Report," per form instructions.

4.1.3 If the correct software revision is available in the database, review the *validation record* (see def.) and planned analysis and computer configuration to determine whether the software has been validated to comply with this procedure; otherwise, go to Step 4.1.4.

4.1.3.1 If the validation applies and is adequate to support the present problem, document the applicability of the existing validation and computer configuration on Form 562.19; otherwise, go to Step 4.1.3.2.

4.1.3.1.1 If the software is installed where it is available for use per this procedure, exit this procedure and do the analysis per MCP-2374, "Analysis and Calculations."

4.1.3.1.2 If the software is not installed where it is available for use per this procedure, contact the *application trustee* (see def.) listed on the Software Inventory Database and obtain access to the software as follows:

4.1.3.1.2.1 Complete Form 562.19 to establish the installation test plan.

4.1.3.1.2.2 Perform the installation test and record results on Form 562.19.

4.1.3.1.2.3 Complete Form 562.20, "Analysis Software Data Input Sheet," and submit to the software librarian.

4.1.3.1.3 Exit this procedure and do the analysis per MCP-2374, "Analysis and Calculations."

4.1.3.2 If the validation does not apply to or is inadequate to support the present problem, complete Section II – Software Activities, of Form 562.19.

4.1.3.2.1 Validate the software per Section 4.2 of this procedure, then exit this procedure and perform the analysis per MCP-2374.

4.1.4 If the analysis software exists and was previously controlled by requirements in MCP-550, but not in accordance with this procedure, evaluate the software per the following steps; otherwise, go to Step 4.1.5

4.1.4.2 Identify the activities to be performed and the documentation required to control the existing analysis software on Form 562.19.

NOTE: *Documentation may include user application requirements, test plans, and cases required to* validate *(see def.) the software for acceptability and user documentation.*

4.1.4.3 Complete Form 562.20.

4.1.4.4 Submit approved Forms 562.19 and 562.20 to the *software librarian* (see def.) for processing in accordance with Section 4.5 of this procedure, then exit this procedure and perform the analysis per MCP-2374.

4.1.5 If the software needs to be developed or modified, go to MCP-550, returning to Step 4.1.4 of this procedure after turnover of the developed/modified software; otherwise, proceed with Step 4.1.6.

4.1.6 If the software is not available in the Software Inventory Database and was not previously controlled by MCP-550, invoke the analysis software controls as follows:

4.1.6.1 Analysis Performer/Manager: Complete Sections I and II of Form 562.19.

4.1.6.2 If software needs to be acquired, obtain the software using the appropriate sections of MCP-1185.

4.1.6.3 Manager: When received, assign technical responsibility for the software application to an application trustee.

4.1.6.4 Application Trustee: Assume technical responsibility for the software application and complete Form 562.19.

4.1.6.5 Record information about the software on Form 562.20, "Analysis Software Data Input Sheet".

4.1.6.6 Submit the completed Form 562.20 supplier-provided *verification documentation* (see def.), supporting documentation, and completed Form 416.04, "Quality Record(s) Validation/Verification Form," to the software librarian for processing.

4.1.6.7 Provide the software librarian with a list of authorized software users.

4.1.6.8 Software Librarian: Update and maintain the library per Section 4.5.

4.2 Validating Software

NOTE: *Software covered by this procedure should not be used until validation activities have been performed and the results approved, recognizing that the final validation can be done in parallel with manager approval as documented on Form 562.19.*

4.2.1 Manager: Assign the task of validating the analysis software to a *validation tester* (see def.).

4.2.2 Manager/Application Trustee: Coordinate with the validation tester and computer support personnel to get the software installed.

4.2.3 Perform an installation test to determine proper installation and integration of the analysis software. Include the following details on Form 562.19, as applicable:

A. Software installation steps to be performed, testing procedures to be used, and criteria for acceptance that installation is adequately completed

B. Controlled document number of the "Installation Test Plan" used

C. If not a controlled document, a note indicating that noncontrolled documentation is attached.

4.2.4 Document the steps taken and reference the supplier's guidance on Form 562.19.

4.2.5 If a warning that existing files will be replaced or changed appears, terminate installation until the effect of the changes on existing validated software can be ascertained.

4.2.6 Validation Tester: Receive requirements from the *manager* (see def.) to validate a specific portion of a software application.

NOTE: *The extent of the validation and the methods chosen are a function of software complexity, degree of standardization, similarity with previously proven software, and the importance to safety. The necessary activities will be documented and approved on the Software Lifecycle Form.*

4.2.7 Obtain documentation from the software librarian to perform validation activities.

4.2.8 Application Trustee: Ensure the validation tester has access to the software.

4.2.9 Validation Tester: Document or update the validation test plan on Form 562.19, ensuring that the test plan will provide adequate validation to support the intended use. Include the following details on the form, as applicable (also see Appendix B for further guidance):

A. Validation testing to be performed, testing procedures to be used, the test problems to be run, and the criteria for validation acceptance

B. Controlled document number of the Validation Test Plan used

C. The following information must be included in the documentation:

- required tests and test sequence
- required ranges of input parameters
- identification of the stages at which testing is required
- criteria for establishing test cases
- identification of test cases that apply to limited installation validation testing
- requirements for test logic branches
- requirements for hardware integration
- anticipated output values
- acceptance criteria
- reports, records, standard formatting, and conventions.

4.2.10 If the documentation is an uncontrolled document, note on the form that noncontrolled documentation is attached.

4.2.11 Software Librarian: Update the software status in the Software Inventory Database to indicate that the software is in the process of validation.

4.2.12 Document the validation test plan acceptance criteria and methods on Form 562.19. Methods that may be used to validate the technical adequacy of the software include without limitation:

A. calculations made without computer assistance (hand calculations).

B. results from other validated software applications.

C. results of experiments and tests.

D. results of standard, confirmed, published problems with known solutions.

E. results of problems independent of the software publisher.

4.2.13 Manager/Application Trustee: Review and approve the validation test planning established on Form 562.19 (the SNF Program must obtain independent review).

4.2.14 Validation Tester: Perform validation tests in accordance with the installation and test plan outlined on the approved Form 562.19. Include the following information:

A. The results of the validation testing performed on the software. If another document was used for this step, indicate the controlled document number, or if not controlled, attach the document to this form and indicate such. Include the following information in this block:

- computer program tested
- computer hardware used
- simulation models used, where applicable
- test problems
- software validation results
- conclusions and resolutions of deficiencies.

4.2.15 Ensure that the software does not perform any unintended functions that, either by itself or in combination with the other functions, can degrade the entire system. (e.g., regression tests, stress tests, integration tests, etc., as applicable).

4.2.16 Specify the hardware and software configuration that applies to the server and/or stand-alone workstation on Form 562.19.

4.2.17 Manager/Application Trustee: Review and approve the results of the installation and validation tests on Form 562.19 (the SNF Program must obtain independent review).

4.2.18 Validation Tester: Submit Form 562.19 to the software librarian (see http://juneau:81/earch/start.html for further information).

4.2.19 Software Librarian: Update the Software Inventory Database with validation documentation, and ensure that the software is linked to Form 562.19, or other validation records.

4.2.20 Submit Form 562.19 to Document Control.

NOTE: *The software is now available for use in the analysis.*

4.3 Reporting Software Errors

4.3.1 Analysis Performer: Report suspected problems in the use of the software to the application trustee, including, if possible, the sequence of events leading up to the suspected problem.

4.3.2 Report limitations and capability differences between versions, anticipated new versions, or other significant information to the application trustee.

4.3.3 Application Trustee: Evaluate the identified problems and determine appropriate action(s) to resolve problems.

NOTE: *The problems are assessed for impact on past and present use of the software. Potential corrective actions include taking the software out of service while the error is being investigated.*

4.3.4 If a defect is identified in the software that adversely impacts previous applications, report the defect per MCP-598.

4.3.5 Communicate the error/problems and corrective actions underway to the software librarian.

4.3.6 Direct the software librarian to notify all software users that an error has been reported.

4.3.7 Software Librarian: Update software status in the Software Inventory Database to include a warning to users about the identified issue(s) and

resolution(s) in progress, and notify software users that an error has been reported.

4.3.8 Analysis Performers: Identify the calculations and analysis results affected by the reported error(s), and evaluate the impact(s).

4.3.9 Perform the necessary evaluations to ensure errors have not affected other calculations and analysis.

4.3.10 Ensure any identified problems and their corrective actions are captured in the MCP-598 process.

4.3.11 Application Trustee: If appropriate, communicate company resolution of action(s) to the software application supplier.

4.3.12 If the software must be modified, make the changes in accordance with MCP-550.

4.3.13 Evaluate error reports sent from the supplier or found on supplier internet home pages for possible impacts to software users.

4.3.14 Inform the software librarian of supplier-reported errors, and ask the software librarian to notify users of the supplier-reported errors.

4.3.15 Software Librarian: Update the Software Inventory Database with a warning to users that the supplier has reported software errors, system patches, workarounds, cautions, warnings and notes or other upgrades, and include a web address to the supplier's internet home page.

4.4 Retiring Software Applications

4.4.1 Application Trustee/Manager: Determine if an analysis software application or a particular version of an application should no longer be used.

4.4.2 Application Trustee: Contact all users of the software application to determine if retiring the software creates a problem.

4.4.2.1 If a software user objects to retiring the software, retain it or resolve the user's concerns and then retire the software.

4.4.3 Communicate the decision to retire the software to the software librarian.

4.4.4 Application Trustee/Manager: If necessary, arrange with computer support personnel to have analysis software retired and removed from the server or a stand-alone workstation.

4.4.5 Ensure the software is removed from the server or the stand-alone workstation.

4.4.6 Inform software librarian that the software has been removed.

4.4.7 Software Librarian: Update the status of the software in the Software Inventory Database as being inactive/removed from use, but ensure historical records are maintained showing the traceability of the unique identification of the software.

4.5 Maintaining and Updating the Library

4.5.1 Software Librarian: Update the Software Inventory Database with information provided by the application trustee/manager.

4.5.2 Ensure that version and/or revision uniquely identify the software.

4.5.3 Maintain the status of the software and a history of any changes to the software.

4.5.4 Maintain a list of authorized software users.

4.5.5 Ensure supplier verification documentation and validation records (approved Form 562.19 or legacy validation documents) are electronically available through EDMS, and referenced to the software it supports.

4.5.6 Receive error notifications from the application trustee and distribute error reports to the user lists, as necessary.

4.5.7 Maintain analysis software under configuration management until the software is retired.

5. RECORDS

Records Description	Uniform File Code	Disposition Authority	Retention Period
Form 562.19, "Software Lifecycle Plan and Report" Form 562.20, "Software Data Input Sheet" Form 416.04, "Record(s) Transmittal" Supplier-provided verification documentation Support documentation for verification and validation of the software	7204	A17-32-a	Retain until structure, system, or component is removed from service.

6. DEFINITIONS

analysis software. Software used in program/project activities to perform analyses or calculations.

application trustee. An individual assigned technical responsibility for an analysis software application. The application trustee acts as the primary point of contact on the software application, provides information on the technical attributes of the software, is responsible for the data associated with the analysis software, works with the software librarian to ensure data is kept current in the Software Inventory Database, is knowledgeable of licensing agreements, is knowledgeable of software users, reviews and approves Form 562.19, resolves software problems and errors, and retires software, when appropriate.

manager. The functional organization manager or the operational manager with budget and technical authority to approve the acquisition/transfer of the software application, approve and fund the Form 562.19, fund the application trustee and the maintenance charges of the software package, and approve the retirement of a software application.

Software Inventory Database. An electronic database located on the INEEL intranet at http://juneau:81/earch/start.html that contains current information about software applications, versions, assigned application trustees, supplier information, validation documentation, and error reports.

software librarian. The individual assigned responsibility for maintaining the Software Inventory Database, controlling the list of software users, receiving error notifications from the application trustee, updating the Software Inventory Database with error reports, and notifying software users of errors, as necessary.

validate. The process of testing software to ensure that it produces expected results.

validation record. A completed Form 562.19 or an existing validation report or plan that documents the planning and test results of validation activities. The validation record is available in the Electronic Document Management System.

validation tester. The individual assigned by the manager with the responsibility for planning, testing, and reporting results of validation activities on analysis software. Software validation is performed by competent individuals, other than those who developed and documented the original design, but who may be from the same organization.

verification documentation. Documents from the supplier that show the software meets the specified requirements.

7. REFERENCES

ASME NQA-1-1997, Subpart 2.7, "Quality Assurance Requirements for Computer Software for Nuclear Facility Applications"

ASME NQA-1-1997, Requirement 3, Section 802, "Software Configuration Management"

DOE/RW-0333P, "Quality Assurance Requirements and Description," Supplement I Software

Form 416.04, "Record(s)Transmittal"

Form 562.19, "Software Lifecycle Plan and Report"

Form 562.20, "Software Data Input Sheet"

MCP-540, "Documenting the Safety Category of Structures, Systems, and Component"

MCP-550, "Software Management"

MCP-598, "Corrective Action System"

MCP-1185, "Acquisition of Materials and Services"

MCP-2374, "Analyses and Calculations"

PDD-122, "Software Quality Assurance Program"

PRD-5092, 19.1, "Software Quality Assurance"

8. APPENDIXES

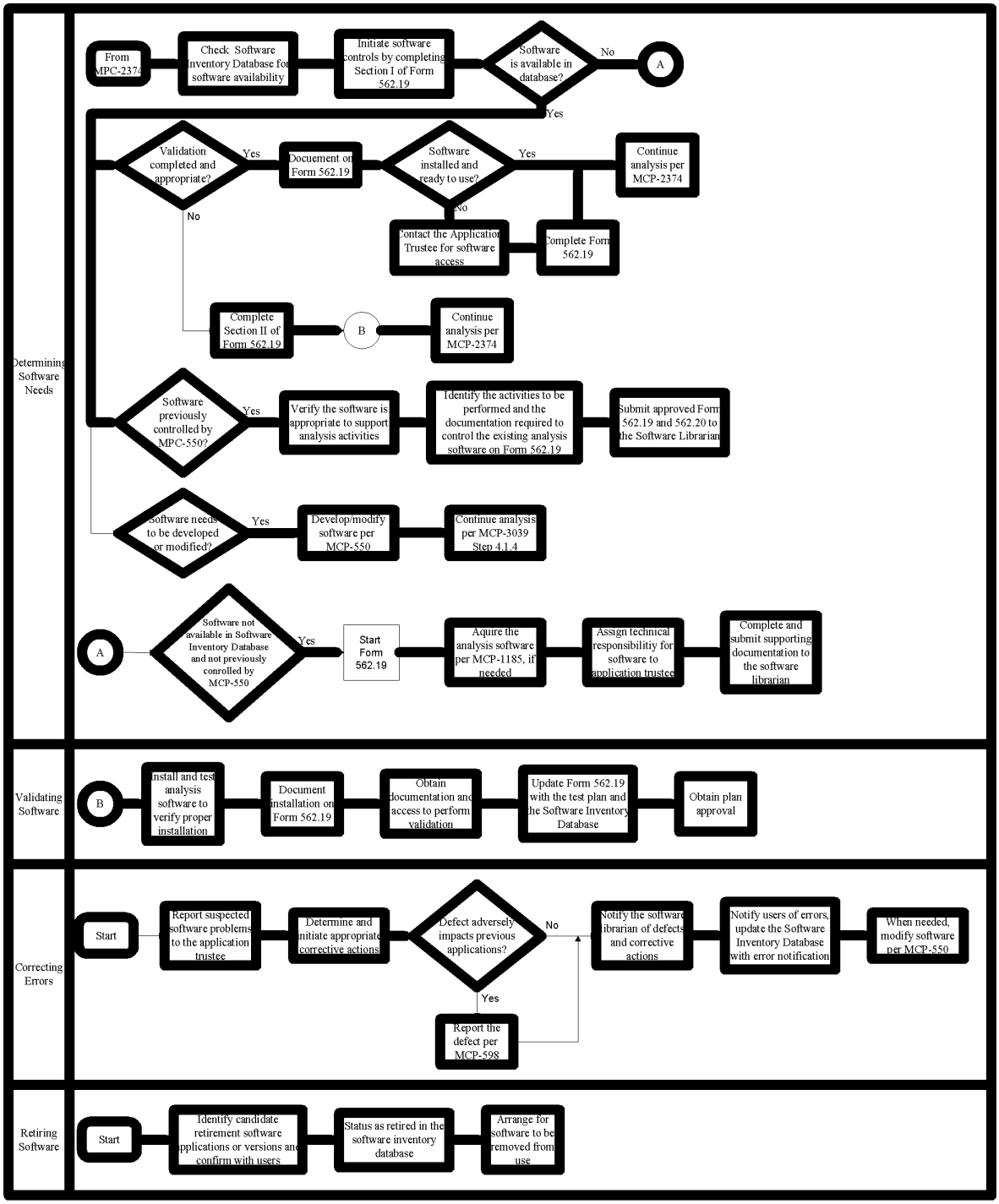
Appendix A, Analysis Software Control Process

Appendix B, Recommended Guidance for Software Installation and Validation Testing Requirements for Modified or New Computer Configurations

Appendix C, Procedure Basis

APPENDIX A

Analysis Software Control Process



APPENDIX B

Recommended Guidance for Software Installation and Validation Testing Requirements for Modified or New Computer Configurations

The Software Lifecycle Plan and Report identifies the computational software (including version) and the computer configuration (specific hardware and operating system software) on which it will be installed. This constitutes a complete installation and validation process for that computational software on the specified computer configuration. This complete process is mandatory, at least initially, for each computational software for which MCP-3039 requires installation and validation testing.

When there is a change to the computer configuration that was approved by the complete installation and validation testing process [e.g., change in operating system software (e.g., software patch), an upgrade in hardware or when the computational software is loaded onto another computer configuration, the following table provides guidance that may be used to assist the application trustee and software tester in determining when the complete installation and validation testing is required, and when limited testing may be sufficient.

Configuration	Configuration Change	Level of Installation and Validation Testing Required
Operating System (assuming no changes in hardware)	Operating system version change (e.g., from Solaris 8 to Solaris 9) or operating system change (e.g., from Solaris to Linux)	Complete[1]
	Modification to the existing operating system affecting computational algorithms (e.g., patch updates that affect math libraries, compilers, etc.)	Limited[2]
	Modification to the existing operating system not affecting computational algorithms (e.g., patches for security, mail utilities, disc drivers, etc.)	No re-testing effort required (previous complete testing still valid)
Hardware (assuming no changes in operating system software)	Identical computer configuration[3]	Limited
	Similar computer configuration[4]	Limited
	Different computer configuration[5]	Complete

1. *Complete*: Complete installation and validation testing is defined by the Software Lifecycle Plan and Report for the specific computational software.

2. *Limited*: Limited installation and validation testing is a subset of the requirements specified in the Software Lifecycle Plan and Report for complete installation and validation testing. For example, if the Software Lifecycle Plan and Report required one installation test problem and ten validation test problems, the limited testing plan may only require the one installation test problem and two of the validation test problems. *The requirements for a limited installation and validation testing plan will be identified on the Software Lifecycle Plan and Report, which would be approved by the application trustee and owning manager.*
3. *Identical computer configuration*: A computer configuration is considered identical to the computer configuration that was previously approved through the complete installation and validation testing process for the computational software if it has, as a minimum, the same CPU model and speed, mainboard, and graphics card. (A variation of this would be, for example, if a Unix machine with 4 CPUs were upgraded to 12 CPUs, using identical boards and CPUs as the original configuration, it would be considered an identical computer configuration as far as installation and validation testing are concerned.)
4. *Similar computer configuration*: A computer configuration is considered similar to the computer configuration that was previously approved through the complete installation and validation testing process for the computational software if it is identical (see above) except for a change in CPU speed (same architecture is required) or minor graphics card change [a minor change is any change (memory, different manufacturer, more capabilities, faster processing , etc.) except going from 2D to 3D capabilities (or vice-versa)].
5. *Different computer configuration*: A computer configuration is considered different to the computer configuration that was previously approved through the complete installation and validation testing process for the computational software if it has a different CPU manufacturer (e.g., Intel vs. AMD), a different CPU architecture (e.g., Intel Zeon vs. Intel Itanium), or a major graphics card difference (2D vs. 3D capabilities).

The above does not address the impact of all available hardware components on computational software validation testing, only those considered “main” components. Given the current technologies in computer hardware, components such as memory, storage (SCSI, IDE), cache, etc., do not appear to affect the accuracy of computational software. This may not be the case in the future. Therefore, changes in these components should be evaluated by the application trustee and software tester with respect to installation and validation testing.

Variation in graphics cards should only be considered as potentially affecting validation to the extent that the validation methods rely upon on-screen geometric representations. If the validation is based upon textual presentation, such as a numerically displayed maximum or minimum, then the graphics card should not be considered a relevant aspect of the hardware configuration for validation purposes.

APPENDIX C

Procedure Basis

Step	Basis	Source Document	Citation
General	Computer software used to produce or manipulate data which is used directly in the design, analysis, and operation of SSCs shall comply with the requirements of this PRD.	PRD-5092, 19.1 Software Quality Assurance PDD-122	4.1.1.1 General
2.0, 1st bullet	Software routines or macros that are documented in each product in which they are used and independently verified by visual inspection or hand calculations without recourse to the originator shall have limited requirements applied….	DOE/RW-0333P	I.2.1.C
4.1.3.2 4.1.4	Existing software and procured or otherwise acquired software that has not been previously approved under a program consistent with the requirements of this document for use in its intended application shall be evaluated in accordance with the requirements of this document.	PRD-5092	4.1.14.1
4.1.3.1	To the extent appropriate, controls shall be established to permit authorized access and prevent unauthorized access to a computer system.	PRD-5092	4.1.15.1
4.1.6.1 and 2	The documentation that is required by this PRD shall be delivered or made available by the supplier to the purchaser. The organization providing software services, such as verification and validation, shall have a plan(s) for software quality assurance that meet the requirements of this PRD as specified in procurement documents.	PRD-5092	4.1.13.2 and 3
4.1.6.6, 4.2.19, 4.3.15, 4.4.7 and 5.	All records designated in implementing documents as quality assurance records shall be controlled in accordance with PRD-5088, 17.1, "Quality Assurance Records"	PRD-5092	4.1.16

Step	Basis	Source Document	Citation
4.2 and Forms 562.19 and 562.20	Validation testing shall be performed in accordance with PRD-5082 and approved test results. Software verification and validation documentation shall describe the tasks and criteria for accomplishing the verification of the software in each phase and the validation of software at the end of the development cycle. The documentation shall: A. Specify the hardware and software configurations pertinent to the software validation B. Be organized in a manner that allows traceability to both software requirements and design C. Contain the results of the execution of the verification and validation activities D. Include the results of the reviews and tests along with the summary of the status of the software	PRD-5092	4.1.7.1–4, 4.1.7.6
4.2.9, 4.2.14 and 4.2.15	Software verification and validation activities shall: A. Ensure that the software adequately and correctly performs all intended functions B. Ensures that the software does not perform any unintended function that either by itself or in combination with other functions can degrade the entire system C. Be planned and performed for each system configuration which may impact the software.	PRD-5092	4.1.2.1

Step	Basis	Source Document	Citation
4.2.14	The results of the verification and validation activities shall be documented with the identification of the verifier indicated.	PRD-5092	4.1.2.3
4.2.18, 4.2.19 and 4.5	Upon acceptable validation of the software, it shall be baselined and placed under configuration management controls.	PRD-5092	4.1.8.1
4.3	A software defect reporting and resolution system shall be implemented for software errors and failures to assure that problems are promptly reported to impacted organizations and to assure formal processing of problem resolutions	PRD-5092	4.1.12.1
4.4	During the retirement phase, the support for a software product is terminated and the routine use of the software shall be prevented	PRD-5092	4.1.10.1
4.5.2 and 4.5.3	Software configuration management includes, but is not limited to, configuration identification, change control, and status control	PRD-5092	4.1.11.1
4.5.7	Configuration items shall be maintained under configuration management until the software is retired.	PRD-5092	4.1.11.2