One Team. One Culture.

Administrative Procedure

PRC-PRO-IRM-309

Controlled Software Management

Revision 5, Change 0

Published: 12/30/2014
Effective: 12/30/2014

Program: Information Technology and Services
Topic: Information Resource Management

Technical Authority: Stewart, Dana
Functional Manager: Johnson, Adria

Use Type: Administrative

JHA: Administrative
Periodic Review Due Date: 12/30/2018
Rev. 5, Chg. 0
USQ Screen Number:

- 100 K Facility : **Screening Determination Performed:**
  USQ 0165-2014
  Screener: Williams, James
- Canister Storage Building/Interim Storage Area : **Categorical Exclusion:** GCX-7 (Minor Change)
  Screener: Covey, Lori
- Central Plateau Surveillance and Maintenance : **Categorical Exclusion:** GCX-7 (Minor Change)
  Screener: Olsen, Ashley
- Less Than HazCat 3 : Excluded from USQ
  Exclusion Reason:
  Exempt from USQ Process per PRC-PRO-NS-062, Scope
- Plutonium Finishing Plant : **Categorical Exclusion:** GCX-7 (Minor Change)
  Screener: King, Jeffry
- Solid Waste Operations Complex : **Categorical Exclusion:** GCX-7 (Minor Change)
  Screener: Olsen, Ashley
- Transportation : Excluded from USQ
  Exclusion Reason:
  N/A per Appendix B, Table B-2
- Waste Encapsulation Storage Facility : **Categorical Exclusion:** GCX-7 (Minor Change)
  Screener: Covey, Lori

CHANGE SUMMARY

Description of Change

Provided option to archive software into alternate approved archival system.
Updated links to informational webpages. Removed Contingency Planning Form form record table.

TABLE OF CONTENTS

1.0 INTRODUCTION ... 3
1.1 Purpose ... 3
1.2 Scope ... 3
1.3 Exemptions ... 3
1.4 Applicability ... 4
1.5 Implementation ... 5
2.0 RESPONSIBILITIES ... 5
2.1 Responsible Manager ... 5
2.2 Software Owner ... 5
2.3 Software Subject Matter Expert (SME) ... 6
2.4 Design/Technical Authority ... 6
2.5 Independent Technical Reviewer(s) ... 6
2.6 Quality Assurance (QA) ... 6
2.7 CHPRC Chief Information Officer (CIO)/Information Management ... 6
2.8 Production Readiness Review Board (PRRB) ... 7
3.0 PROCESS ... 8
3.1 Software Management Methodology ... 9
3.1.1 Establish Roles and Responsibilities for Software Management ... 10
3.1.2 Software Identification, Registration, and Grade Level Determination ... 11
3.1.3 Financial Risk Management ... 17
3.1.4 Minimum Software Controls ... 19
3.1.5 Software Project Planning ... 20
3.1.6 Software Management and Quality Planning ... 23
3.1.7 Software Requirements Identification and Management ... 26
3.2 Software Acquisition ... 29
3.3 Software Development ... 32
3.3.1 Software Design ... 32
3.3.2 Software Implementation ... 34
3.4 Software Testing ... 36
3.5 Approval for Use ... 39
3.6 Software Installation and Checkout ... 41
3.7 Lessons Learned ... 44
3.8 Software Operation ... 44
3.9 Software Maintenance ... 47
3.10 Software Retirement ... 49
3.11 Approval and Issuance of Software Documentation ... 51
4.0 FORMS ... 52
4.1 Format Guidance ... 52
5.0 RECORD IDENTIFICATION ... 53
6.0 SOURCES ... 54
6.1 Requirements ... 54
6.2 References ... 54

List of Appendixes

Appendix A - Hanford Information System Inventory (HISI) Software Grading Checklist... 55
Appendix B - Software Grading Criteria 59
Appendix C - Glossary 68

1.0 INTRODUCTION

1.1 Purpose

This procedure establishes the methods to manage software and the software lifecycle activities, which include requirements identification, configuration control, acquisition, evaluation, development, design, implementation, verification and validation, operation and maintenance (O&M), and retirement of software.

1.2 Scope

This procedure is applicable to CH2M HILL Plateau Remediation Company (CHPRC) employees and major subcontractors involved in Plateau Remediation Contract (PRC) scope of work. The *Software Grading Checklist (SGC)* (Appendix A -), found in the Hanford Information Systems Inventory (HISI), is used to determine how to apply this procedure to PRC software. This procedure applies to:

- Nuclear and safety-related software including safety system software, safety analysis, and design software, software that performs safety-related functions and distributed control systems (DCS)
- Quality-affecting software used to design, characterize, or verify quality of analyses and work processes, including compliance with environmental permits and other regulatory requirements
- High and moderate-risk business applications that support PRC work scope

This procedure describes a graded approach to be applied to the software lifecycle. The *Software Grading Criteria* (Appendix B -) is used to determine the level of control and the required lifecycle phase documentation based on the software category. The rationale for the grading criteria is to provide differing levels of reviews, testing activities, and documentation based on the safety-significance of the software, regulatory requirements, quality function, and the risk of software failure.

NOTE: *Terms specific to this document are defined in* Appendix C - *, Glossary*.

1.3 Exemptions

NOTE: *U.S. Department of Energy (DOE) "toolbox" codes and any software applications that have been mandated by the DOE for PRC use are not exempt from this procedure, as documentation must be established based on the application of the code to the work scope.*

The following types of software are exempt from the requirements specified in this procedure:

1. Support software and system software (Microsoft Windows®) provided as part of the Hanford site computer image managed by the Information Resources Management (IRM) service provider, or other support/system software that does **not** generate data or perform process control functions that could affect the software.

2. Site Licensed software managed by the IRM service provider (see listings in Software Distribution, "Desktop Applications and Internet Tools" and Electronic Software Acquisition.)

NOTE: *Desktop software such as Microsoft Excel® and Microsoft Access®, as software tools are themselves exempt, but applications built using them are not (refer to Section 3.1.2.1).*

3. "Flat-file" spreadsheets that are wholly incorporated into technical reports, calc-notes, or other documentation where the calculations, mathematical formulas, and input data can be exactly verified during the technical review of the report.

4. Equipment or components that have integrated software in which the source code is not alterable by the user and where functionality of the system is demonstrated through Acceptance or Operational Tests, or calibration over the operational range. Systems with embedded software or firmware are managed in accordance with PRC-RD-EN-1819, *CHPRC Engineering Requirements.* This does not apply for control system and data acquisition software for physical plant systems (see #6 below).

5. Low risk, standalone, desktop tools that are used only for efficiency (i.e., PDF Converter). These tools are **not** used to calculate or generate data used to make a company decision and will **not** influence or impact the company or project mission.

The following types of software are only required to follow this procedure through the completion of HISI registration and the SGC for software acquisition or development (refer to Section 3.1.2.3). After completion of the HISI entry, the following software is controlled by the procedure indicated:

6. Software used for engineering analysis, data acquisition, design activities, or control software for physical plant systems, is managed in accordance with PRC-PRO-EN-20050, *Engineering Configuration Management* and PRC-PRO-EN-40357, *Engineering Software Management.*

7. Legacy Software placed into "standby" status. Steps required to place software into standby status are found in Section 3.1.2.5 of this procedure.

1.4 Applicability

This procedure encompasses the entire software lifecycle and applies to PRC software and the personnel that develop, procure, modify, test, maintain, operate, use, or retire the software.

1.5 Implementation

This procedure is effective upon publication. Implementation of any changes in this procedure is expected upon the next update to software currently managed by this procedure.

The software owner is responsible for implementing this procedure. The documentation required and the rigor of the implementation process is determined by the software grade level.

2.0 RESPONSIBILITIES

2.1 Responsible Manager

- Assigns technically competent Software Subject Matter Expert(s) (SME) for the organization, project, function, or facility, as applicable
- Assigns technically competent Software Owner(s) for the organization, project, function, or facility, as applicable
- Assigns technically competent software developers and/or analysts to manage, develop, implement, and maintain software application(s) for the organization, project, function, or facility, as applicable
- Reviews and approves software documentation, if required
- Considers the effects implementing the new software will have on state and local government regulations and consult with other agencies, as appropriate

2.2 Software Owner

- Is knowledgeable in the work processes and software used in their organization
- Is knowledgeable in the nature of the information, process, management, personnel, operation, security, and technical controls of the application
- Is responsible for the performance and control of individual software applications, completion of software documentation, and management and training of authorized users
- Has sufficient time and effort to ensure that procedural compliance is achieved and maintained
- Ensures implementation of applicable software lifecycle requirements
- Completes the HISI SGC to determine software grade and applicable software controls in accordance with this procedure
- Is responsible for identifying risks and planning adequate security, taking into account the security of all environments in which the application will operate

- Assigns technically competent software test personnel and independent technical reviewers for the software, as applicable
- Ensures maintenance of current information about the software application in HISI, including links/locations to the latest revision of all available documentation
- Ensures software baseline records (including media, documentation, code, etc.) are managed in accordance with PRC-PRO-IRM-10588*, Records Management Processes*
- Reviews and approves software documentation produced by others

2.3 Software Subject Matter Expert (SME)

- Acts as the expert on software management requirements contained within this procedure
- Has collective cognizance of software applications in use within the organization, project, function, or facility, as applicable
- Ensures correct and complete data entry of software information in HISI
- Reviews and approves HISI entries on SGCs and Version Description tabs
- Reviews and approves software documentation

2.4 Design/Technical Authority

- Provides input and approves requirements, acquisition, design, configuration management, testing, maintenance, and retirement work activities and documentation for safety software and quality affecting software applications, as appropriate

2.5 Independent Technical Reviewer(s)

- Reviews requirements, design, code, testing, maintenance work activities, and documentation for software applications, and provides feedback/comments to the software owner, as appropriate

2.6 Quality Assurance (QA)

- Provides software quality oversight throughout the software lifecycle
- Is assigned at the project, program, and/or facility level

2.7 CHPRC Chief Information Officer (CIO)/Information Management

- Approves all software registrations in HISI
- Updates HISI when software is retired or system owner identified in HISI is no longer with the company

- Provides assistance in data preservation to ensure data is retrievable in its native format once a system has been archived and removed from production
- Ensures the IRM service provider is provided access to vendor documentation through the vendor, vendor's website, or CHPRC access-controlled share areas, as appropriate
- Ensures the IRM service provider manages support software, system software, and site-standard software in accordance with a Quality Assurance (QA) program in compliance with CRD O 414.1D, *Quality Assurance*
- Ensures the IRM service provider has a documented procedure for handling emergency software changes that cannot be implemented as part of a scheduled release

2.8 Production Readiness Review Board (PRRB)

- Reviews and approves applications' readiness for installation on the Hanford Local Area Network (HLAN)

3.0 PROCESS

The sequence of the process sections in this procedure are in the order that will most likely be followed in the development and acquisition of new software, or in updating legacy software. When appropriate, some processes may be conducted in parallel.

This procedure specifies requirements for multiple documents; these requirements can be met using separate documents or can be combined into one or more consolidated documents. If documents are combined or consolidated the method for such should be described within the affected documents and/or the HISI Version Description tab fields for the affected documents.

Format guidance is referenced throughout this procedure as a tool for use in meeting documentation requirements. Format guidance may be combined, used individually, or modified, provided that the content requirements are met. For format guidance, refer to the Software Document Templates webpage located on the IT&S website. This webpage also provides links to checklists that are used to assess software documentation for compliance to requirements, which can be used as a guide in preparing software documentation.

Responsibilities are assigned to specific job titles. The persons with these job titles are responsible for ensuring that the work is completed. For some projects the job title may be shared by more than one person.

Section 3.1 Software Management Methodology – assign resources, and plan software acquisition and/or development

Section 3.2 Software Acquisition – acquire software from a commercial supplier, government code center, or internet download (i.e. freeware/shareware)

Section 3.3 Software Development – develop software through establishing requirements, performing design, and implementing code

Section 3.4 Software Testing – plan and perform testing of software to ensure that the software fully meets the specified requirements

Section 3.5 Approval for Use – review and approve a software product and documenting the formal release of the software for use

Section 3.6 Software Installation and Checkout – install software on operating computer platforms to ensure proper performance in the user environment

Section 3.7 Lessons Learned – document problems or successes that are incurred during the implementation and installation phases

Section 3.8 Software Operation – operate software in accordance with specified user limitations to ensure safe and reliable operation of the software

Section 3.9 Software Maintenance – identify and report software problems, and take corrective action to modify software, to resolve errors, or to add new enhancements to meet changing application requirements

Section 3.10 Software Retirement – retire software from operation when it is no longer needed

Section 3.11 Approval and Issuance of Software Documentation - obtain approvals and issue software documents and/or place electronic records into the Integrated Document Management System (IDMS)

3.1 Software Management Methodology

The software management methodology involves the application of a systematic, disciplined, and quantifiable approach at each stage of the software lifecycle. This method starts with defining conceptual models of a software application or system and using these models as the basis for system specification and design. In addition, the method establishes standards for planning the work, developing the design, coding, testing, and writing user documentation. It is imperative to verify the completion of each lifecycle stage, and to control changes to the baseline configuration.

Software management should include the following principles:

- Consider, at each stage of the information lifecycle, the effects of decisions and actions on other stages of the lifecycle, particularly those concerning information dissemination.
- Seek to satisfy new information needs through interagency or intergovernmental sharing of information, or through commercial sources, where appropriate, before creating or collecting new information.
- Integrate planning for information systems, plans for resource allocation and use, including budgeting, acquisition, and use of information technology.
- Protect government information commensurate with the risk and magnitude of harm that could result from the loss, misuse, or unauthorized access to or modification of such information in accordance with PRC-PRO-IRM-184, *Information Protection and Clearance*.
- Develop information systems that facilitate interoperability, application portability, and scalability of applications across networks of heterogeneous hardware, software, and telecommunications platforms.
- Record, preserve, and make accessible sufficient information to ensure the management and accountability of agency programs, and to protect the legal and financial rights of the Federal Government.

3.1.1 Establish Roles and Responsibilities for Software Management

3.1.1.1 Assignment of Software SME(s)

Actionee	*Step*	*Action*
Responsible Manager	1.	ASSIGN an employee to act as the Software SME for the organization, project, function, or facility.
Software SME Candidate	2.	COMPLETE online Software SME training (Course No. 600005).
	3.	COMPLETE the Software SME request for approval through the Software SME application (http://www7.rl.gov/ssme/). a. ENTER the two (2) -digit department code of the organization, project, or facility to be supported. b. IF a Software SME will be supporting more than one (1) organization, THEN ENTER each organization code that will be supported.
Responsible Manager	4.	INDICATE approval in the Software SME application for the candidate Software SME to be assigned for the organization, project, function, or facility, as applicable.
Department Manager	5.	INDICATE approval in the Software SME application for the candidate Software SME to be assigned for the organization, project, function, or facility, as applicable.

3.1.1.2 Assignment of Software Owners

Actionee	Step	Action
NOTE:		*Software owners should be chosen and assigned based not only on their qualifications and experience in the field that the software covers, but also on their roles and responsibilities within the PRC. The Software Owner's primary responsibilities should reflect the level of commitment necessary to provide adequate management of the software, and should not hinder the implementation of this procedure.*
Responsible Manager	1.	ASSIGN an employee to act as the Software Owner for software application(s).
Software Owner	2.	IDENTIFY AND ASSIGN software development project team members.
Software Project Team (including Owner)	3.	COMPLETE online Controlled Software Management training (Course No. 600006).

3.1.2 Software Identification, Registration, and Grade Level Determination

At the beginning of each software project, the software must be registered in the HISI, and the type and grade level of the software must be identified.

Careful considerations should be made in identifying the correct control requirements for Microsoft SharePoint® and spreadsheet calculation software, as directed below.

3.1.2.1 SharePoint Applications

Microsoft SharePoint® can be used to build software applications that can be reused. This section applies to applications built using SharePoint®. SharePoint® applications, sites, or sub-sites shall be registered in HISI if any of the following criteria is met.

Actionee	Step	Action
Software Owner or	1.	DETERMINE if the SharePoint® site or sub-site is considered software:

Actionee	*Step*	*Action*
Responsible Manager		a. IF **any** of the following criteria are met, the controls of this procedure apply, THEN PROCEED with Section 3.1.2.3 • A web application has been created to support the creation or management of one (1) or more SharePoint® site collections, applications, or sites • Custom software applications created using SharePoint® functionality • Customized code has been applied to create or alter any web part or workflow included in the site or sub-site • An interface has been created to receive or send data to another software application or site/sub-site b. Calculations that may be considered Utility Calculation Software (or a critical spreadsheet) when a calculation has been created using lists or libraries in a site/sub-site. Refer to Section 3.1.2.2 for additional information. c. IF **none** of the above criteria are met, THEN EXIT this procedure and control the SharePoint® site in accordance with management direction.

3.1.2.2 Spreadsheets and Utility Calculation Software

Utility calculation software tools, such as Microsoft Excel® and MathCAD, can either be used as desktop calculators, or to build software applications that can be reused. This section applies to applications built using utility calculation tools. Applications developed with these tools will be categorized as Utility Calculation software. If the utility calculation tool is only used as a desktop calculator, the requirements of this procedure do not apply, since a software application is not created.

Actionee	*Step*	*Action*
Software Owner or Responsible Manager	1.	DETERMINE if the spreadsheet is **not** considered Utility Calculation software and is governed by another procedure: a. IF **all** of the following criteria are met, the controls of this procedure do **not** apply, THEN EXIT this procedure AND CONTROL the software application in accordance with the procedure that governs the calculation. • It does not contain any macros or programming • All inputs, formulas, and output data are reviewed during the

Actionee	*Step*	*Action*
		checking of the calculation • Each calculation is reviewed and signed • The review is required by a procedure b. IF **three (3) or less** of the above criteria are met, THEN PROCEED with Section 3.1.2.3.

3.1.2.3 Software Application Identification

This section involves identifying that the software has been registered in HISI and that the SGC has been approved.

Actionee	*Step*	*Action*
Software Owner	1.	IF the software is already in use within the PRC, THEN DETERMINE if the software has been registered in HISI and the SGC has been approved: a. IF the software has been registered and the SGC has been approved, THEN PROCEED to Section 3.1.2.4, step 5. b. IF the software has been registered but the SGC has **not** been approved, THEN REVIEW AND CORRECT the SGC as needed THEN PROCEED to Section 3.1.2.4, step 3.
Software Owner		c. IF the software has **not** been registered in HISI, THEN CONTINUE with Section 3.1.2.4, step 1.

3.1.2.4 Software Registration and Grading

Planning a software project begins with registering the software in HISI and completing the SGC to determine the software grade, which then dictates the applicable set of controls and documentation requirements.

Actionee	*Step*	*Action*
NOTE:	*HISI registrations for control system and data acquisition software for Configuration Managed Structures, Systems, and Components (SSCs) engineering analysis and design activities, and control software for physical plant systems are completed in accordance with PRC-PRO-EN-20050 and PRC-PRO-EN-40357.*	
Software Owner	1.	REGISTER the software by completing the HISI registration form: http://msc.ms.rl.gov/hisi/form/register.cfm.
		a. IF the software is known to be governed by another procedure, THEN ENTER the procedure number and title in the Comments field.
	2.	COMPLETE the SGC in HISI. CONSIDER system interfaces, planned usage, data sensitivity, contingency of operation, intended use, and impacts.
		a. ANSWER “yes” or “no” to each question in relation to how the software will be used. SEE Appendix A - for additional details.
		b. SELECT a Software SME that is assigned to the organization, project, function, or facility that manages the software.
		c. INITIATE the approval cycle of the SGC.
Software SME	3.	REVIEW the HISI registration and SGC entry for correctness and completeness.
		a. IF the entry is approved, THEN INDICATE approval on the HISI SGC tab, AND CONTINUE with step 4.
		b. IF the entry is **not** approved, THEN CONTACT the software owner for corrections AND RETURN to step 1.

Actionee	*Step*	*Action*
CHPRC CIO	4.	REVIEW the HISI registration and SGC entry for correctness and completeness. a. IF entry is approved, THEN INDICATE approval on the HISI SGC tab, AND CONTINUE with step 5. b. IF entry is **not** approved, THEN CONTACT the software owner for corrections AND RETURN to step 1.
Software Owner	5.	Based on the type, grade level, and use of the software, PROCEED with software evaluation, planning, and implementation: a. IF the software will be used for engineering analysis and design activities and control software for physical plant systems, THEN EXIT this procedure AND MANAGE the software in accordance with PRC-PRO-EN-40357. b. IF the software is Utility Calculation software which performs design and safety-related calculations that affect SSCs activities, THEN ENSURE the software is developed and managed in accordance with PRC-PRO-EN-40357, AND CONTINUE with Section 3.1.2.5. c. IF the software is legacy software, and is level A-E, THEN CONTINUE with Section 3.1.2.5. d. IF the software is a new development or existing software to be accepted for another use, and is level A-E, THEN CONTINUE with Section 3.1.3. e. IF the software is considered low risk (no-level or N/A), THEN EXIT this procedure AND CONTROL the software using a graded approach in accordance with management direction.

3.1.2.5 Legacy Software Evaluation and Planning

Legacy software applications and systems must be identified and controlled as part of the CHPRC inventory. Legacy software is held to the same requirements as non-legacy software, and must be managed in accordance with this procedure. On an interim basis, a Justification for Continued Use of Software can be utilized until compliance is achieved, as defined in the process steps below.

Actionee	*Step*	*Action*
Software Owner	1.	DETERMINE if the legacy software has the appropriate software documentation in accordance with this procedure and Appendix B - and if the software has been approved for use. a. IF the software is currently in use and documentation and approvals are adequate, AND: 1) IF the software is grade A, B, C, or D, THEN PROCEED to Section 3.8 to continue operation. 2) IF the software is grade E, THEN PROCEED to Section 3.1.3, step 8 to continue operation. b. IF the software has been placed in “standby” status due to its relation to a facility or project lifecycle stage, THEN: 1) ENSURE the HISI Version Description tab includes correct and complete information, including document number, title, and location/links to latest revision of all available documentation listed. 2) IF the graded approach was used to combine documents, THEN DESCRIBE the graded implementation in applicable Version Description tab fields. 3) UPDATE the HISI Core Information Status to “On Standby.” 4) MAINTAIN the software baseline in the current configuration until: a) IF the facility life or project is completed or canceled, THEN RETIRE the software in accordance with Section 3.10. b) IF the software has been identified to return to service, THEN CHANGE the HISI status back to “In Development” AND IMPLEMENT a new version of the software in accordance with Section 3.1.4.

Actionee	*Step*	*Action*
Software Owner		c. IF the software is currently in use and documentation, testing, or approvals are **not** adequate, THEN: 1) GATHER any historical information or Software Quality Assurance (SQA) documentation related to the software. 2) ENTER a description of the information gathered, including links to existing documentation if available, in the HISI Version Description tab. 3) IF no information or documentation was discovered for a particular item, THEN ENTER a description of the results of the investigation in the Version Description tab. 4) PERFORM an evaluation to determine if the software meets existing documented software requirements and determine adequacy of existing design, test, and user documentation to support software O&M. 5) REPORT inadequacies discovered in the Condition Reporting and Resolution System (CRRS) in accordance with PRC-PRO-QA-052, *Issues Management*.

NOTE: *Corrective actions, extended due dates, intended length of continued use of software, and relation of software to the facility lifecycle should be clearly described on the form.*

6) PREPARE a *CHPRC Justification for Continued Use of Software* form (A-6006-161 or equivalent).

NOTE: *The link to the approved Justification of Continued Use form should be entered in the HISI Core Information "Comments" or Version Description "Other Documents" fields.*

7) OBTAIN approvals
AND ISSUE *CHPRC Justification for Continued Use of Software* in accordance with Section 3.11,
THEN CONTINUE operation:

a) IF the software is grade A, B, C, or D,
THEN PROCEED to Section 3.8 to continue operation.

b) IF the software is grade E,
THEN PROCEED to Section 3.1.4, step 8 to continue operation.

3.1.3 Financial Risk Management

This phase addresses the financial aspects related to planning the software project, including performing an Alternatives Analysis, with considerations given to the risks that may be involved.

Actionee	*Step*	*Action*
Software Owner	1.	EVALUATE the financial risk of the software application based on the projected software lifecycle cost. a. IF the modification, acquisition, or development lifecycle cost is expected to exceed $250,000 the application will be considered a large-scale installation, THEN CONTINUE to step 2. b. IF the modification, acquisition, or development lifecycle cost will **not** exceed $250,000, THEN DOCUMENT results of evaluation in the Alternatives Analysis A field on the HISI Version Description tab, AND PROCEED based on the following grade level: 1) IF level E, THEN CONTINUE with Section 3.1.4. 2) IF level A, B, C, or D, THEN CONTINUE with Section 3.1.5.

NOTE: *The choice to develop a customized application should be balanced against the availability of other solutions; and the project cost, resources, and time constraints.*

2. PERFORM an Alternatives Analysis.

- DOCUMENT the alternative acquisition or development methods considered for reducing lifecycle costs in an Alternatives Analysis report (format guidance - or equivalent)
  AND INCLUDE a description of the following:
- The proposed system functions, requirements, and constraints
- Viable alternatives, related assumptions, and dependencies, including efforts to meet development needs from the following methods:
  - Adapt an existing software application from within the Hanford Site
  - Adapt an existing software application from other government agencies
  - Purchase Commercial-Off-the Shelf (COTS)software application
  - Reusing existing modules of code
  - New custom developed software application

Software Owner

- Alternatives ranking and selection description (discuss any redundancies or reduction of redundancies)
- Cost justification and return of investment (include cost savings, increased quality, speed, flexibility, customer satisfaction, etc.)

Actionee	Step	Action
	3.	PREPARE a *Disaster Recovery Plan* (format guidance or equivalent).
	4.	OBTAIN approvals AND ISSUE the *Alternatives Analysis Report* and *Disaster Recovery Plan* in accordance with Section 3.11, THEN PROCEED based on grade level: a. IF level E, THEN CONTINUE with Section 3.1.4 b. IF level A, B, C, or D, THEN CONTINUE with Section 3.1.5.

3.1.4 Minimum Software Controls

This section applies to grade level E non-safety software.

Actionee	Step	Action
Software Owner	1.	DOCUMENT the functional requirements of the new or updated software application in a *Functional Requirements Document* (FRD) (format guidance - or equivalent).
	2.	IF the software application's data or database contains controlled-use information, THEN ENSURE the FRD includes requirements to mark and protect data, database, and output in accordance with PRC-PRO-IRM-184.
	3.	IF the application contains controlled-use information, THEN COMPLETE OR REVISE a *Data Security Plan* (format guidance – or equivalent) that defines, at a minimum: • data protection • data distribution/usage • access controls • encryption • uses • output
Software Owner	4.	IF the software is custom developed, THEN PREPARE a *Software Design Description* (SDD) (format guidance - or equivalent) that documents the overall structure (control and data flow) and the reduction of the overall structure into physical solutions (e.g., algorithms, equations, control logic, and data structures).
Programmer	5.	DEVELOP AND INSTALL the software application.

Actionee	Step	Action
Test Personnel	6.	TEST the software application to verify that it performs correctly, documenting the results on an *Acceptance Test Report* (ATR) (format guidance - or equivalent).
Software Owner	7.	ISSUE the FRD, Data Security Plan, SDD, and ATR in accordance with Section 3.11, THEN CONTINUE with step 8.
User(s)	8.	OPERATE software: a. IF a change is needed, THEN NOTIFY the software owner AND RETURN to step 1.
Software Owner	9.	IF software is no longer needed, THEN RETIRE the software in accordance with Section 3.10.

3.1.5 Software Project Planning

Software project planning should be performed before the development, acquisition, or major modification of a software application.

Actionee	Step	Action
Software Owner	1.	DETERMINE the type of software implementation project to be planned: a. IF the software has been identified to be used for a new purpose or scope of work, THEN PROCEED to Section 3.1.5.1. b. IF the software is a new implementation or a major modification to existing software, THEN PROCEED to Section 3.1.5.2.

3.1.5.1 Acceptance for New Scope

This phase consists of defining and documenting the basis for accepting existing, approved, software baselines for new scope or application. For example, although CFAST (Consolidated Fire and Smoke Transport) is capable of both thermal and smoke transport for fire safety analysis, it may have only been validated for thermal transport. This section establishes the method for evaluating the existing software application for the new intended usage and accepting the software for the new application scope.

Actionee	Step	Action
Software Owner	1.	For existing software that has been identified to be used for a new purpose, PREPARE OR UPDATE the FRD (format guidance - or equivalent) defining:

Actionee	Step	Action
		• Critical features • Capabilities • Security requirements • Interfaces to be included in the required software application
		2. IF the software application's data or database contains controlled-use information, THEN ENSURE the FRD includes requirements to mark and protect data, database, and output in accordance with PRC-PRO-IRM-184.
		3. ISSUE the FRD in accordance with Section 3.11, THEN CONTINUE with step 4.
		4. EVALUATE the existing software application to determine if the current software design, test, and user documentation supports O&M for the new scope per risk and graded approach as outlined in Appendix B - , *Software Grading Criteria*, THEN CONTINUE with step 5.
		a. IF the software is not adequate to support user application requirements, THEN MODIFY software in accordance with Section 3.1.5.2.
		b. IF the SRS or SDD do not address the intended use, THEN MODIFY software documentation in accordance with Section 3.3.
		c. IF the existing test plans, test cases, and test reports are not adequate to validate the software for its intended use, THEN PERFORM acceptance testing in accordance with Section 3.4.
Software Owner		d. IF the existing user documentation does not comply with the minimum requirements of Section 3.3.2, THEN PREPARE *User Documentation* (format guidance - or equivalent).
		5. SUBMIT the software application for approval for use in accordance with Section 3.5.

3.1.5.2 New Implementation or Major Modification to Existing Software

This section involves defining and documenting the basis for a new software application or a major modification to software that is already in use.

Actionee	*Step*	*Action*
Software Owner	1.	For new software, or major modifications to existing software applications, PREPARE OR UPDATE the FRD (format guidance - or equivalent) defining: • Intended application at Hanford • Critical features • Capabilities • Interfaces to be included in the required software application • Security requirements for application code, access, and data

NOTE: *Software applications should operate without workstation local administrator privileges. Documented justification may be required to acquire or develop an application that requires privilege.*

2. IF the software application's data or database contains controlled-use information,
   THEN ENSURE the FRD includes requirements to mark and protect data, database, and output in accordance with PRC-PRO-IRM-184.
3. For Level A/B/C software applications or new technology, SUBMIT the FRD to the CHPRC CIO's Office for a cyber-security and new technology review and approval.
4. OBTAIN approvals
   AND ISSUE the FRD in accordance with Section 3.11
   THEN CONTINUE with Section 3.1.6.

3.1.6 Software Management and Quality Planning

Software management and quality planning is performed before the development, acquisition, or major modification of a software application. A Software Management Plan (SMP) is required to be written to define specific requirements, procedures, or methods for a particular software application or organization. SMPs must be reviewed and updated, as necessary, prior to the development, acquisition, or major modification of an application.

This procedure specifies requirements for multiple documents; these requirements can be met using separate documents or can be combined into one or more consolidated documents as appropriate in accordance with Appendix B - . If documents are combined or consolidated, the method for such should be described within the affected documents and/or the HISI Version Description tab fields for the affected documents.

Actionee	*Step*	*Action*
NOTE:		*If project size or complexity warrants separation of some documents from the SMP, the owner may utilize templates for a Software Configuration Management (SCM) Plan (format guidance or equivalent) or Software Training Plan (format guidance or equivalent). Separate document links should be placed in the HISI Version Description tab in the related sections (i.e., Training Plan link placed in the User Training section.)*

Software Owner	1.	PREPARE OR UPDATE the SMP (format guidance - or equivalent), to contain the following information:
		• A description of the overall nature and purpose of the software application(s)
		• Organizational structure
		  ○ Identify organizations responsible for performing the work
		  ○ Identify organizations responsible for achieving software application quality
		  ○ Organizational tasks and responsibilities
		• Software development and acquisition
		  ○ Software engineering method
		  – Explanation or description of quantifiable and measurable controls used during design
		  ○ Design basis for the software application
		  – Standards, conventions, techniques, or methodologies that shall guide code development
		  ○ Supporting documentation to be prepared and maintained

Actionee	*Step*	*Action*
Software Owner		○ Intended method for graded approach toward document preparation – Required software application reviews – Documentation method ○ Verification and validation methods • Configuration management methodology, including the following, as applicable: ○ Methods used to control, uniquely identify, describe, and document the configuration version or update of software and its related documentation should be documented ○ Criteria for configuration identification, change control, configuration status accounting, and configuration reviews and audits • Risk Management Risks can pertain to items such as user input errors, system stability, data output security, financial/budgetary impacts, support concerns, resource issues, scheduling, managing users, version controls, dependence on outside sources, or changes in requirements. ○ Risk Management Techniques – Techniques planned to control risks throughout the lifecycle ○ Risk Identification – Listing of technical and managerial project risks that have been identified, assessed, and prioritized ○ Risk Mitigation – Description of options planned for risk avoidance, mitigation, or transfer options • Software operation and maintenance ○ Error reporting and corrective action methods ○ Software application controls that address the security systems for the protection of software and data

Actionee	*Step*	*Action*

Software Owner		• Data management
		  ○ Methods for maintaining the integrity of data, mathematical models, and output files
		  ○ Method for managing controlled-use and manage in accordance with PRC-PRO-IRM-184, including non-disclosure agreements with third parties
		  ○ Plan for maintaining data security, including access, encryption, interfaces, uses, and output
		• Training Plan
		  ○ Training planned for analysts, development team, and test team in fault tolerant, safety design, user interface design, testing methodologies, and configuration management to ensure delivery of a robust software application.
		  ○ Training planned for users and operations staff in proper data entry, use of option and menus, and correct interpretation of software results.

	2.	For all interfaces where data is shared, PREPARE *Non-Disclosure Agreements* for all interfaces, using Form A-6005-194, *CHPRC Controlled-Use Information Nondisclosure Agreement*, or Form A-6005-193, *Third-Party Nondisclosure Agreement*, as appropriate.
	3.	OBTAIN approval and ISSUE the SMP and forms in accordance with Section 3.11, and THEN CONTINUE with Section 3.1.7.

3.1.7 Software Requirements Identification and Management

This phase consists of defining and documenting the requirements the software application must satisfy, including the response of the software to anticipated classes of input data and the detail and information necessary to acquire or design the software.

Actionee	*Step*	*Action*
Software Owner	1.	PREPARE a *Software Requirements Specification* (SRS) document (format guidance - or equivalent) that defines the requirements the software must satisfy, including the response of the software to anticipated classes of input data and the detail and information necessary to design the software. In the SRS, INCLUDE the requirements for: • Functionality - the functions the software is to perform • Performance - the time-related issues of software operation such as speed, recovery time, response time, etc. • Design constraints imposed on implementation phase activities - any elements that will restrict design options • Attributes - one-time-related issues of software operation such as portability, acceptance criteria, access control, maintainability, etc. • External Interfaces - interactions with people or external systems • Software Controls that address the security systems and user for the protection of software and data in a manner that is consistent with its threats and missions • Identification system if it contains controlled-use information • Safety system related requirements, as applicable • Security features (vulnerability protection, cyber security, etc.)
Software Owner or Responsible Manager	2.	ASSIGN an Independent Technical Reviewer who did not work on the original software development or modification, but may be from the same organization.

NOTE: *Documented completion of the review may be indicated through the approval of the Independent Technical Reviewer on the document. Comments from the review not incorporated into the current document must be retained until they are incorporated.*

Actionee	*Step*	*Action*
Independent Technical Reviewer	3.	REVIEW the SRS to ensure that the requirements are complete, verifiable, consistent, and technically achievable. This review shall also ensure that the requirements will result in feasible and usable code.
Software Owner	4.	OBTAIN approvals AND ISSUE SRS in accordance with Section 3.11, THEN CONTINUE with step 5.

	5.	PREPARE AND INITIATE the use of a *Requirements Traceability Matrix* (RTM) (format guidance - or equivalent). The purpose of the RTM is to trace software requirements through definition (provided in the FRD and SRS), design (provided in the SDD), and testing (provided in test plans, test cases, and the ATR). SEE Section 4.1, *Format Guidance* for acronym descriptions and template links. The RTM, or equivalent, should include the following information, as applicable:
		
		• Unique Identification Number(s): The number assigned to the requirement from the requirements document(s) (FRD, SRS, and SMP).
		
		  ○ IF a unique number was not assigned to each requirement prior to creating the RTM,
		  THEN CREATE a unique identification number for each requirement, such as RTM-001, RTM-002, etc., to identify requirements within the RTM.
		
		• Requirement(s): Description of the requirement as it relates to the software product
		
		• Source of Requirement: The source from which the software requirement originated. This can come from a number of sources, such as DOE regulations, configuration control boards, technical authorities, procedure manuals, customer requirement, business case, etc.

Actionee	*Step*	*Action*
		• Requirements Document/Section and ID: Reference to the requirements document(s) where the requirement is defined. Provide the document number (if known), section reference where the requirement is described, and a unique identifier for the requirement, as applicable. ○ IF the requirement source is driven by a contract, THEN CITE the current contract and revision number. – Requirements are usually defined in the FRD or the SRS. – Requirements may also be defined in a SMP. • Design Specification: The design document where the requirement is defined. PROVIDE the document number, section reference where the requirement is described, AND unique identifier for the requirement, as applicable. This field is not applicable for acquired software.
Software Owner		• Program Module: For modular-developed software systems, the name or identifier of the software where the requirements is implemented. • Test Case Number(s): Reference to the test case number and document where the requirement test case is defined. INCLUDE the document name, document number (if known), test case number, and any other pertinent identifying information, such as section reference. Test cases are usually defined in the *Software Test Plan* (STP) and the ATR. • Successful Test Verification: Indication of the completion of software testing for the requirement verifying the software's capability to meet the criteria for success as defined by the requirements. • Modification of Requirement: Description of whether the requirement was changed, eliminated, or replaced, along with the disposition and authority for modification. • Remarks: Any pertinent remarks or comments relating to the requirements.
	6.	ENSURE that the RTM is updated, as applicable, throughout the design and testing phases.

Actionee	Step	Action
		a. WHEN these phases and the RTM are complete, THEN OBTAIN approvals AND ISSUE the RTM in accordance with Section 3.11.
	7.	PROCEED with selected acquisition or development method, in accordance with the SMP, SRS, and appropriate related forms/documentation.
		a. IF the software application involves acquisition of software or software services, THEN INITIATE the procurement in accordance with Section 3.2.
		b. IF the application involves CHPRC development of new software, or major modifications to existing software, THEN PROCEED with Section 3.3.

3.2 Software Acquisition

Software and services that use software can be procured through several means including P-cards, Statements of Work (SOW), and work for others. For further information on the acquisition process, refer to PRC-PRO-AC-40500, *Acquisition Management Plan*.

Actionee	Step	Action
NOTE:	•	*In cases when DOE mandates the use of a particular software application, or DOE "toolbox" codes are used, the acquisition process still applies.*
	•	*All software acquisitions made from non-Hanford sources, including freeware and shareware, are required to be documented and approved on a P-Card ADP Approval form.*
Software Owner	1.	DETERMINE the acquisition method to be used:
		a. PURCHASE software using a P-Card for items defined within limits identified in PRC-PRO-AC-335, *Use and Control of Purchasing Card*; PROCEED with step 5.

Actionee	Step	Action
		b. PURCHASE the following items/services using the contract/purchase order process; CONTINUE with step 2. • Custom software or software application • Instrumentation and Control (I&C) software • Services which use software • Software maintenance services

NOTE: *Software SME and CHPRC CIO approval of software maintenance services is not required.*

2. INCLUDE the SME and CHPRC CIO as reviewers of the SOW for acquisition of software application or software services.

NOTE:
- *The SOW should flow down software QA and configuration management requirements, particularly in the area of services where software will be used. Procurement, contracts, and QA staff may be contacted to assist with clauses and standardized language pertaining to the use and protection of data and software maintenance are is included.*
- *Safety control system software shall be purchased and managed in accordance with PRC-PRO-EN-40357.*

Software SME	3.	REVIEW AND APPROVE the SOW for acquisition of the software application or software services.
CHPRC CIO	4.	IF the application is Level A/B/C, THEN REVIEW AND APPROVE the SOW for acquisition of software application or services.
Software Owner	5.	NOTIFY the CHPRC Information Management organization for review and approval of procurement of the new software or Information Systems technology.

NOTE: *Purchased or acquired software may fall under Federal Copyright laws. Software license management requirements and responsibilities are defined in the "CHPRC Code of Business Ethics and Conduct." It is recommended that a legal review of any associated license agreement be performed to ensure an understanding of all clauses and conditions.*

6. COMPLETE purchase of the software application or services using the contract/purchase order process identified above.

Actionee	*Step*	*Action*
Software Owner	7.	Upon approval, VIRUS-SCAN any files or software downloaded from the Internet or received from any other outside source before opening it in accordance with PRC-PRO-IRM-592, *Unclassified Computer Security*.
	8.	ESTABLISH the software baseline by collecting existing software and supporting documentation (e.g., code disks/CD-ROM).

NOTE: *Software documentation provided by the vendor may be issued as a CHPRC software record in accordance with* Section *3.11.*

	9.	ENSURE all documents specified in the procurement contract have been provided by the vendor.
	10.	MANAGE the software code disks/CD-ROM in accordance with the SMP, ensuring that the disk copies are stored in an area/facility other than where the application operates, AND MEET record requirements, if applicable.
	11.	ARCHIVE the completed software code in either of the Centralized Software Configuration Management (SCM) Systems (i.e., MKS Source Integrity or Microsoft Visual Source Safe®) or a CHPRC CIO approved alternate archival system.
Software Owner	12.	REVIEW software baseline as recorded in HISI to determine the adequacy of the software documentation to support O&M per risk, and graded approach as outlined in Appendix B - , *Software Grading Criteria*. • IF existing documentation is not adequate, THEN PREPARE additional required documents, or obtain additional documentation from the vendor. • IF the existing test plans, test cases and test reports provided by the vendor are not adequate to evaluate the software for its intended use, THEN DEVELOP additional testing requirements.
	13.	For services that use software, VERIFY the following: • Software used in the service is identified • The vendor's software QA program and submitted documentation meet requirements specified in the SOW

3.3 Software Development

Software development is conducted by first delineating specific requirements for the software application, developing a design from the requirements, and implementing the design by writing program code.

3.3.1 Software Design

The design phase involves translating software requirements into a documented software design. Results of the design may necessitate modifications to the requirements specification.

Actionee	*Step*	*Action*
Software Owner or designee	1.	PREPARE a SDD (format guidance - or equivalent) that documents the overall structure (control and data flow) and the reduction of the overall structure into physical solutions (e.g., algorithms, equations, control logic, and data structures) AND INCLUDE in the *SDD* the following information, as applicable: • A description of the major components of the software design as they relate to the software requirements • Software safety implementation, including identification of hazard mitigation strategies • A technical description of the software application with respect to the theoretical basis, mathematical model, control flow, data flow, control logic, and data structures

Actionee	*Step*	*Action*

Software Owner or designee		• A description of the allowable or prescribed ranges for inputs and outputs
		• List of Integration Points (e.g., HLAN interfaces)
		• Data Model (Metadata)
		  ○ Incorporate disposition instructions for the data, including records management
		  ○ Explain the physical and technical characteristics of data records, including a record layout that describes each field including:
		  – Name
		  – Size
		  – Data form (such as alphabetic, zoned decimal, packed decimal, or numeric), or a data dictionary
		  – Description of the relationship between data elements in data bases
		  – Other technical information needed to read or process the records
		• Hardware and Software Configuration.

	2.	UPDATE the RTM, tracing the design elements to the software requirements.
NOTE:		*Documented completion of the review may be indicated through the approval of the Independent Technical Reviewer on the document. Comments from the review not incorporated into the current document must be retained until they are incorporated.*
Independent Technical Reviewer	3.	REVIEW the SDD to assure that the design is complete, verifiable, consistent, and traceable to the functional and performance requirements specified in the SRS.
Software Owner	4.	OBTAIN approvals <u>AND</u> ISSUE SDD, in accordance with Section 3.11, <u>THEN</u> CONTINUE with Section 3.3.2.

3.3.2 Software Implementation

The implementation phases consist of translating the software design into executable code. The requirements, as described in the SRS, and the design, as described in the SDD, may need to be modified to reflect coding solutions developed during the implementation phase.

Actionee	*Step*	*Action*
Software Owner	1.	ASSIGN a software developer to create the source code based on the design.
Software Developer	2.	CREATE the source code based on the design, AND ANALYZE AND DEBUG the code to identify AND CORRECT errors.
	3.	DOCUMENT the code, user-generated parameters, scripts, and metadata required to compile and generate the executable program. This documentation may be provided as code listings or on electronic media.
Independent Technical Reviewer	4.	PERFORM a code walkthrough or inspection, including but not limited to review to reduce complexity of source code. a. DOCUMENT the code walkthrough in a letter report or other method (e.g., verification checklist).

Actionee	*Step*	*Action*
Software Owner	5.	PREPARE *User Documentation* (format guidance - or equivalent) AND INCLUDE the following: • Software name and version identifier • Description of the user's interaction with the software • Description of any training necessary to use the software • Input and output specifications • Input and output formats • Description of software and hardware limitations • Minimum user hardware and software configurations • Information for obtaining user and maintenance support • Description of user messages initiated as a result of improper input and how the user can respond • Process for reporting problems, changes, or obtaining help
Software Owner	6.	OBTAIN approvals AND ISSUE the code walkthrough and user documentation in accordance with Section 3.11, THEN CONTINUE with step 7.

NOTE: *The centralized configuration management system is the tool used to manage software baseline items that are designated to become electronic records until they are transferred to IDMS or converted to paper records for transfer to the Records Holding Area (RHA).*

7. ESTABLISH the software baseline by collecting existing software and supporting documentation (e.g., code disks/CD-ROM) AND COMPLETE the following:
   - MANAGE the software code disks/CD-ROM in accordance with SMP ensuring disk copies are stored in an area/facility other than where the application operates.
   - ARCHIVE the completed software code in either of the Centralized SCM Systems (i.e., MKS Source Integrity or Microsoft Visual Source Safe®) or a CHPRC CIO approved alternate archival system.

Actionee	Step	Action
	8.	TEST the implemented software application in accordance with Section 3.4.

3.4 Software Testing

The testing phase is a formal validation of the code through manual or automated execution of the test cases presented in the STPto demonstrate that the software satisfies specified requirements, properly integrates with hardware systems, and produces correct results. Acceptance criteria shall be used to identify differences between expected and actual results.

Actionee	Step	Action
Software Owner	1.	ASSIGN test personnel within the following constraints: • Test personnel cannot test portions of the software that they have developed. • Test personnel may include the Software Owner and the Software SME.
	2.	ASSIGN required independent technical reviewers that are not software developers nor test personnel.
Software Owner	3.	IF the software is I&C software, THEN ENSURE equipment is tested in accordance with PRC-PRO-EN-40357 ensuring that the test plan and test report documentation meet the minimum requirements as described in steps 4 and 5.

Actionee	Step	Action

	4.	DOCUMENT test requirements by preparing a STP (format guidance - or equivalent). Include in the test plan the following, as applicable:
		• Software identification (name and version)
		• System Software used
		• Test requirements and technical approach
		  ○ For developed software applications
		  – Design-based testing requirements
		  – Robustness
		  – Test of all design requirements
		  ○ For acquired software applications, including freeware/shareware and DOE “toolbox” codes:
		  – Acceptance testing requirements specified in procurement documents
		  – Supplemental testing requirements for acquired software applications
		  – Demonstration of capabilities within limitations for its intended use
		  ○ Required ranges of input parameters
		  ○ Requirements for testing logic branches
		  ○ Requirements for hardware integration
		  ○ Requirements for unintended function test
		  ○ Requirements for demonstrating the proper functioning of documented protective measures
		• Evaluation method(s) to be used
		• Sequence of required activities
		• Acceptance criteria for review of test results

Actionee	*Step*	*Action*
Software Owner	5.	OBTAIN approvals AND ISSUE the STP in accordance with Section 3.11, AND CONTINUE with step 6.

NOTE: *Testing of acquired software must demonstrate that it will operate successfully within the limitations of its intended application on site.*

Software Testers	6.	PERFORM testing in accordance with the approved STP.
Software Owner	7.	REPORT software problems identified during testing, including resolution and retesting if completed during testing, by preparing a *CHPRC Problem Report/Change Request form* (PR/CR) (A-6005-146 or equivalent) in accordance with Section 3.9.
	8.	PREPARE an ATR (format guidance - or equivalent) to document acceptable performance of the software. Include the following information in the ATR: • Description of, or reference to, the applicable test plan, test cases, test requirements, and acceptance criteria • Sufficient documentation to prove implementation of protective measures is consistent with the specifications • Documentation of its performance without work station local administrator privileges • Sufficient documentation to demonstrate acceptable performance • Description of any changes in evaluation methods, inputs, or test sequence • Date of test • Simulation models used, when applicable • Hardware used and its configuration during the test • Test results and conclusions concerning acceptance and utilization of the software • Identification and signature of test personnel
	9.	UPDATE the RTM tracing design elements and requirements to test elements.

Actionee	*Step*	*Action*
Software Owner or Responsible Manager	10.	ASSIGN an Independent Technical Reviewer who did not work on the original software development or modification but may be from the same organization.

NOTE: *Documented completion of the review may be indicated through the inclusion of results and approval of the Independent Technical Reviewer on the appropriate document(s). Comments from the review not incorporated into the current documentation must be retained until they are incorporated.*

Independent Technical Reviewer	11.	REVIEW the STP, ATR, and RTM for completeness, consistency, clarity and correctness. Acceptable test methods may include: • Comparison with hand calculations • Comparison with calculations using comparable proven problems • Comparison with information from confirmed published data • Comparisons with other approved computer programs • Comparisons with experiments and tests • Selected retesting of acceptance tests, as appropriate • Manual inspections of system performance
Software Owner	12.	OBTAIN approvals AND ISSUE the ATR and RTM in accordance with Section 3.11, THEN CONTINUE with Section 3.5.

3.5 Approval for Use

This phase involves obtaining the necessary approvals to install software applications.

Actionee	*Step*	*Action*
Software Owner	1.	INITIATE a review of the software, including the following documents, in accordance with PRC-PRO-NS-062, *Unreviewed Safety Question Process*. • HISI SGC • SMP • SDD • STP and Report a. PROVIDE a link to related review documentation with a description of the results, or describe the results of the review if no document was provided, in the “Other Documents” field of the Version Description tab.

NOTE: *The issuance of software documentation and electronic records processing may*

Actionee	Step	Action
		take time to be completed. Approval for use of software may be granted in HISI prior to final issuance and electronic record processing, when appropriate. In such cases, links to issued documents and electronic records of software documentation supporting the software release should be added to the HISI entry within 30 days following implementation.
Software Developer/ Analyst/ Project Lead	2.	ENSURE each field of the Version Description tab has been completed, as appropriate. AND INCLUDE the applicable document number, title, and location/link to each relevant document for the current software version to be installed. • IF the graded approach was used to combine documents, THEN DESCRIBE the graded implementation in the applicable Version Description tab fields.
	3.	SELECT “Submit” on the HISI Version Description tab to initiate the automated approval process.
Software Owner	4.	REVIEW the HISI Version Description tab for completeness, consistency, and adequacy. a. IF the entry is approved, indicate your approval on the HISI Version Description tab, THEN CONTINUE with step 5. b. IF the entry is not approved, THEN CONTACT the software developer for corrections AND RETURN to step 2.
Software SME	5.	REVIEW the HISI Version Description tab for completeness, consistency, and adequacy. • IF the entry is approved, THEN INDICATE your approval on the HISI Version Description tab, AND CONTINUE with step 6. • IF the entry is not approved, THEN CONTACT the software developer for corrections AND RETURN to step 2.

Actionee	*Step*	*Action*
CHPRC CIO	6.	REVIEW the HISI Version Description tab for completeness, consistency, and adequacy. • IF the entry is approved, THEN INDICATE your approval on the HISI Version Description tab, AND CONTINUE with Section 3.6. • IF the entry is not approved, THEN CONTACT the software developer for corrections AND RETURN to step 2.

3.6 Software Installation and Checkout

The installation phase is the period in the software lifecycle when the product is integrated into its operational environment and tested to ensure that it performs as required. Checkout testing demonstrates that all key performance functions (e.g., logon/logoff, interfaces, links, screens, and reports) perform as expected.

The PRRB conducts reviews of software applications to be installed on HLAN to ensure that the installation will not interfere with HLAN's operation or schedules, and to coordinate with other groups and contractors.

Actionee	*Step*	*Action*

NOTE: *Generally, HLAN based applications meet one of the following criteria:*

- *Software and data will reside on an HLAN server*
- *Software will control data transfers between HLAN servers*
- *Software will be made available via Software Distribution*

*The following are **not** considered HLAN based:*

- *No interfaces with external data or applications*
- *Desktop application installed on a group share area and only pulling data from the same group share area*
- *Standard software interfaces with the HLAN which are controlled through system software, such as network printing and file management*

Actionee	*Step*	*Action*
Software Owner	1.	PRRB review and approval is required for all applications to be installed on HLAN (identified in HISI's core section, "Base" field, as "Network – HLAN"). PRRB approval is indicated on the HISI Version Description tab. a. IF PRRB review is required, THEN CONTINUE with step 2. b. IF PRRB review is **not** required, THEN PROCEED to step 7.

NOTE: *In determining the level of review required, consider the following:*

- *Minor: User interface changes, report changes, limited number of users affected, bug fix, etc.*
- *Major: New functionality, database structure changes, changes in business process or procedure*
- *Emergency: Loss of service, security upgrades, etc.*

Software Owner	2.	DETERMINE the level of review required: a. IF the software release is "minor," THEN CONTINUE with step 6. b. IF the software release is "emergency," THEN CONTINUE with step 3. c. IF the software release is "major," or is determined to have a major impact to the HLAN, THEN CONTINUE with step 4.
PRRB	3.	DETERMINE if "emergency" software release requires a follow-up presentation with the PRRB. a. IF emergency change is considered major, THEN CONTACT the software owner (or delegate) to schedule a presentation date AND CONTINUE with step 4. b. IF emergency change is considered minor, THEN CONTINUE with step 6.

Actionee	*Step*	*Action*
	4.	DETERMINE if an in-person presentation to the weekly PRRB meeting is required to describe the planned software release AND ANSWER questions to HLAN support organization representatives. a. IF an in-person PRRB presentation is required, THEN CONTACT the software owner (or delegate) to schedule presentation date, AND CONTINUE with step 5. b. IF an in-person PRRB presentation is **not** required, THEN PROCEED to step 6.
Software Owner or Delegate	5.	PRESENT information to the PRRB regarding the software release to be installed. This presentation will be based on the HISI Version Description tab entries, but may include additional details, as requested.
PRRB	6.	REVIEW AND APPROVE PRRB submittal in the Version Description section of HISI, approving installation.
Software Owner or Designee	7.	INSTALL software on computer platform.
Software Owner	8.	ASSIGN test personnel ensuring they only test portions of the software they have not developed.
Test Personnel	9.	DOCUMENT results of installation tests on the *CHPRC Software Installation & Checkout Form* (A-6005-149 – or equivalent).
Software Owner	10.	Once the software installation and checkout has been performed, and problems are successfully resolved, OBTAIN approvals AND ISSUE the *CHPRC Software Installation & Checkout Form* in accordance with Section 3.11, THEN CONTINUE with step 11.
	11.	UPDATE the HISI Core Information status to “Operational,” as applicable.
	12.	ARCHIVE the completed software in either of the Centralized SCM Systems (i.e., MKS Source Integrity, Microsoft Visual Source Safe®) or CHPRC CIO approved alternate archival system.

3.7 Lessons Learned

The lessons learned phase relates to documenting problems or successes incurred during the implementation and installation phases of the software.

Actionee	Step	Action
NOTE:		*Upon completion of the post-implementation information on the HISI Version Description tab, an automated record is transferred to the IDMS capturing the approval for use of the current software version.*
Software Owner	1.	DOCUMENT a summary of the implementation of the software in the post implementation section of the HISI Version Description tab. Note the following: • A brief description of any issues and their corrections • If there were no issues with the implementation of the software

3.8 Software Operation

The operation phase is the period in the software lifecycle when the application is in operation, in accordance with the software requirements, design, and the SMP.

Actionee	Step	Action
Software Owner	1.	INITIATE user passwords, access controls, and system vulnerability protections, as applicable.
	2.	PROVIDE required user training in accordance with the following criteria: a. DOCUMENT user qualifications to minimize the risk of improper or invalid use of the software application. b. DOCUMENT user training to minimize the risk of improper or invalid use of the software application.
	3.	DEVELOP <u>AND</u> MANAGE an *Authorized User List*. • For Level A/B/C safety analysis and design software, users must be authorized based on their training and qualifications. ENTER user names individually into the HISI Authorized User tab (or equivalent). • For other software applications, the *Authorized User List* may be controlled using the HISI Authorized User tab, or referenced in HISI, as appropriate.

Actionee	Step	Action
NOTE:		• *Usage logs may be maintained in HISI to rapidly assess the impact of a software application error on work done using the application.* • *Logging usage is **not** required for:* ○ *I&C software installed in facility systems/components/instruments and operated in accordance with approved procedures.* ○ *Applications which are used for routine operations where usage is recorded within the application or in related operating documentation.* ○ *Usage data is covered by Unclassified controlled nuclear information (UCIN) or Official-use only (OUO) restrictions.*
	4.	LOG Level A/B/C software usage in HISI, completing the HISI Usage Log tab (or equivalent), as appropriate. a. COMPLETE the usage log fields with sufficient data to show historical use of the software.
Software Users	5.	OPERATE software in accordance with user documentation.
Software Owner	6.	IF level A/B/C, THEN PERFORM periodic operational testing where computer failure or electronic drift can affect safety systems. AND DOCUMENT operational testing in an ATR (format guidance - or equivalent, i.e., *Facility Modification Package* (FMP) or Work Package). a. IF operational testing indicates hardware or system change, THEN INITIATE a CHPRC *Problem Report/Change Request (*PR/CR) *form* (A-6005-146 or equivalent) in accordance with Section 3.9.
	7.	PERFORM a periodic review, at least annually, of the HISI contacts, core information, and the SGC, AND UPDATE information as necessary.
	8.	CONFIRM the HISI information review by clicking the “Owner Review Completed” button in the core information section in HISI.

Actionee	*Step*	*Action*
	9.	Periodically REVIEW the SMP and *CHPRC Justification for Continued Use of Software* to determine if a revision is needed. Revisions may be required when: • A review of the document has not been performed in two (2) years • A major organizational change has occurred • Software development, acquisition, operations, or maintenance methods have changed significantly • Acquiring, developing, or retiring a major system a. IF a revision to the document is not needed, THEN DOCUMENT the results of the review in the SMP field of the HISI Version Description tab, THEN PROCEED with step 10. b. IF a revision is required, THEN REVISE the SMP in accordance with Section 3.1.6 or the J*ustification of Continued Use of Software* in accordance with Section 3.1.2.4, AND CONTINUE managing the software in accordance with the SMP or *CHPRC Justification of Continued Use of Software*.
Software Owner	10.	CONTINUE software operation until a change is identified: a. IF a software problem or change is identified, THEN PERFORM maintenance in accordance with Section 3.9. b. IF the software is no longer needed, THEN RETIRE the software in accordance with Section 3.10.

3.9 Software Maintenance

The maintenance phase relates to the period in the software lifecycle when the application is in operation; changes may be requested and problems may arise.

Actionee	*Step*	*Action*
Software User	1.	NOTIFY the Software Owner of software application problems or required software application changes. a. IF a code change is required to address a software application problem or error, THEN DECIDE whether to continue using the software application, AND INITIATE a problem evaluation in accordance with step 2. b. IF a code change is required to address new requirements, or upgrades to the software application, THEN INITIATE change request in accordance with step 5.
Software Owner	2.	CONTACT Computer Technical Support for network system support, or the software vendor, as needed to determine if the problem is software related. a. IF the problem is software related and requires vendor modification, THEN DOCUMENT requested change per vendor contract AND PROCEED with acquisition as a new version. b. IF the problem is software application related, THEN evaluate the problem, AND CONTINUE with step 3. c. IF the problem is ***not*** software-related, THEN CONTACT the Software SME to determine required action to resolve the identified problem.
	3.	INVESTIGATE the impact of the problem on previously reported data or affected hardware/software.
Software Owner	4.	IF the problem was determined to be a software application error that was reported or used in the performance of work, THEN REPORT the error to the appropriate level of management in accordance with PRC-PRO-QA-052.

Actionee	*Step*	*Action*
	5.	COMPLETE a *CHPRC Problem Report/Change Request* (PR/CR) *form* (A-6005-146 or equivalent) AND DOCUMENT the following, as applicable: • Nature of required change (Minor/Major) • Disposition and Proposed Corrective Action • Impact Analysis • Affected Hardware components • Affected Baseline Documentation • Affected software (e.g., libraries, databases) • Testing Requirements
Software SME	6.	REVIEW AND APPROVE the completed PR/CR AND RETURN the PR/CR to the software owner to implement the change.
Software Owner	7.	DETERMINE if the required change to the software application will affect functional and performance requirements. a. IF functional or performance requirements are affected, this constitutes a major change THEN PROCEED with software development in accordance with Section 3.3. b. IF a minor change is required, THEN CONTINUE to step 8.
Programmer	8.	IMPLEMENT the approved software application change AND UPDATE affected documentation as needed.
Test Personnel	9.	DEFINE test cases required to verify the minor change and to support regression testing.
	10.	PERFORM required testing of software in accordance with the testing requirements specified on the PR/CR. DOCUMENT test results on the PR/CR or an ATR.
Independent Technical Reviewer	11.	For Level A/B/C software applications, REVIEW test results for completeness and adequacy.

Actionee	*Step*	*Action*
Software SME	12.	REVIEW AND APPROVE the PR/CR test documentation as required for completeness, consistency, and adequacy.
Software Owner	13.	OBTAIN approvals AND ISSUE the PR/CR in accordance with Section 3.11, THEN CONTINUE with step 14.
	14.	LABEL each configuration item in such a manner that the identification method uniquely identifies each approved configuration of the software that is available for use, AND ARCHIVE in either of the Centralized SCM Systems (i.e., MKS Source Integrity, Microsoft Visual Source Safe®), or CHPRC CIO approved alternate archival system.
	15.	SUBMIT the software application for approval for use in accordance Section 3.5.

3.10 Software Retirement

The retirement phase in the software lifecycle is when it has been determined that the software is no longer needed, and that retiring the software is appropriate. This determination should reflect projected needs as well as required support for past and current work.

Actionee	*Step*	*Action*
Software Owner	1.	DEVELOP criteria to determine when the software can be retired from use and the methods to prevent the use of retired software.
	2.	REQUEST to retire the software by completing a CHPRC *Software Retirement Checklist* (A-6005-151 or equivalent).
	3.	OBTAIN approvals AND ISSUE the *Software Retirement Checklist* in accordance with Section 3.11, THEN CONTINUE with step 4.
	4.	ENSURE the HISI Version Description tab includes correct and complete information regarding the final software version to be retired, including links to the latest revision of all documentation listed.
	5.	OBTAIN approvals for retirement on the Version Description tab in accordance with Section 3.5, AND CONTINUE with step 6.

Actionee	*Step*	*Action*
Software Owner	6.	REMOVE retired software from production, AND ARCHIVE the software version, AND TERMINATE support in accordance with the Retirement checklist.
	7.	ENSURE the software baseline is archived as electronic record data in accordance with PRC-PRO-IRM-10588 AND the following requirements: • For archiving in electronic format: ○ ARCHIVE data files and databases in a software and hardware independent format, such as flat files, rectangular tables, two-dimensional arrays, or lists. ○ All data records should have the same logical format. ○ Each data element within a record should contain only one data value. ○ A record should not contain nested repeating groups of data items. ○ The file should not contain extraneous control characters, except record length indicators for variable length records, or marks delimiting a data element, field, record, or file. ○ If records or data elements in different files need to be linked or combined, then each record must contain keys enabling linkages between the related records in separate files. ○ Archive electronic documents as plain ASCII files; electronic documents may contain Hypertext Markup Language (HTML) tags.
	8.	UPDATE the HISI Core Information Status to “Retired.”

3.11 Approval and Issuance of Software Documentation

Actionee	*Step*	*Action*
Software Owner	1.	IDENTIFY appropriate reviewers for the document: a. For level A/B/C software, ENSURE the Facility Design/Technical Authority reviews and approves documentation related to software specifications, acquisitions, design and development, verification, testing, software configuration management, maintenance, and retirement.
NOTE:		*QA reviewers may be identified for selected software documents to facilitate QA oversight throughout the lifecycle, otherwise QA oversight may be performed in accordance with* PRC-MP-QA-599*.*
Software Owner		b. INCLUDE additional reviewers as identified on forms or directed by management, as applicable.
Software Owner/SME	2.	At a minimum, APPROVE any document that has no other approvers identified.
Software Owner	3.	DETERMINE the type of software documentation to be approved and issued. a. IF the document is a report type document that should be controlled, requires an information clearance review, or may be revised during the software lifecycle (i.e., SMP or SDD), THEN CONTINUE with step 4. b. IF the document is a form or other document that is prepared specifically for a one-time use (i.e., code walkthrough), THEN CONTINUE with step 5.
	4.	FINALIZE preparation of document, OBTAIN approvals, AND ISSUE document in accordance with PRC-PRO-IRM-9679, *Administrative and Technical (Non-Engineering) Document Control,* THEN CONTINUE with step 9.
	5.	ENSURE the document is easily identifiable including a header or a coversheet that includes the following, at a minimum: • Software Acronym and Name • Document Type or Title (i.e., Code Walkthrough) • Document Author Name and Company • Document Number, if applicable. If no number has been assigned "N/A" may be used • Document Date
	6.	OBTAIN approvals on the document, or include other evidence of approval (i.e., email approval) with the document.

Actionee	*Step*	*Action*
Software Owner	7.	PREPARE a *CHPRC Active Records Transmittal* (form A-6005-152), identifying the document and its document number, software acronym, and document type/title, as applicable.
	8.	PROVIDE the document and *Active Records Transmittal* form to the IT&S electronic records point of contact.
IRM Service Provider	9.	PROCESS the document as an electronic record into the IDMS.
	10.	SEND an email confirmation of completion of processing including the IDMS link to the document to the software owner.
Software Developer/ Analyst/ Project Lead	11.	ENTER the document number, document title, and the IDMS link to the document into the appropriate field of the HISI Version Description tab. a. IF no field matches the document type/title, THEN USE the "Other Documents" field.

4.0 FORMS

CHPRC Active Records Transmittal, A-6005-152

CHPRC Controlled-Use Information Nondisclosure Agreement, A-6005-194

CHPRC Justification for Continued Use of Software, A-6006-161 (or equivalent)

CHPRC Problem Report/Change Request Form (PR/CR), A-6005-146 (or equivalent)

CHPRC Software Retirement Checklist, A-6005-151 (or equivalent)

CHPRC Software Installation and Checkout Form, A-6005-149 (or equivalent)

Third-Party Nondisclosure Agreement, A-6005-193

4.1 Format Guidance

Format guidance templates are found on the Software Document Template page of the IT&S website.

Functional Requirements Document (FRD)

Alternatives Analysis

Software Management Plan (SMP)

Software Requirements Specification (SRS)

Software Design Description (SDD)

Software Test Plan (STP)

Acceptance Test Report (ATR)

User Documentation

Data Security Plan

Disaster Recovery Plan

Requirements Traceability Matrix (RTM)

5.0 RECORD IDENTIFICATION

All records are generated, processed, and maintained in accordance with PRC-PRO-IRM-10588.

Records Capture Table

Name of Record	Submittal Responsibility	Retention Responsibility
Functional Requirements Document	Software Owner	IRM Service Provider
Software Management Plan	Software Owner	IRM Service Provider
Software Baseline Code	Software Owner	IRM Service Provider
Acceptance Test Report	Software Owner	IRM Service Provider
Authorized User List	Software Owner	IRM Service Provider
Software Design Description	Software Owner	IRM Service Provider
Alternatives Analysis Report	Software Owner	IRM Service Provider
Software Requirements Specification	Software Owner	IRM Service Provider
Code Walkthrough	Software Owner	IRM Service Provider
User Documentation	Software Owner	IRM Service Provider
Test Plan	Software Owner	IRM Service Provider
Disaster Recovery Plan	Software Owner	IRM Service Provider
Data Security Plan	Software Owner	IRM Service Provider
CHPRC Justification for Continued Use of Software A-6006-161 (or equivalent)	Software Owner	IRM Service Provider
CHPRC Software Installation & Checkout Form A-6005-149 (or equivalent)	Software Owner	IRM Service Provider
CHPRC Software *Retirement Checklist* A-6005-151 (or equivalent)	Software Owner	IRM Service Provider
CHPRC Problem Report/Change Request form (PR/CR) A-6005-146 (or equivalent)	Software Owner	IRM Service Provider
CHPRC Controlled-Use Information Nondisclosure Agreement, A-6005-194	Software Owner	IRM Service Provider
Third-Party Nondisclosure Agreement, A-6005-193	Software Owner	IRM Service Provider
Requirements Traceability Matrix	Software Owner	IRM Service Provider

6.0 SOURCES

6.1 Requirements

10 CFR 830, *Nuclear Safety Management Subpart A – Quality Assurance Requirements*

36 CFR 1234, *Electronic Records Management*

CRD O 200.1A, *Information Technology Management*

CRD O 205.1B, *DOE Cyber Security Management Program*

CRD O 414.1D, *Quality Assurance*

CRD O 415.1, *Information Technology Project Management*

CRD O 420.1C, *Facility Safety*

DOE/RL-96-68, Rev 3, *Hanford Analytical Services Quality Assurance Requirements Document*

NQA-1-2008, *Quality Assurance Requirements for Nuclear Facility Applications*

NQA-1a-2009 addenda, *Quality Assurance Requirements for Nuclear Facility Applications*

OMB Circular A-130, *Management of Federal Information Resources*

PRC-MP-QA-599, *Quality Assurance Program*

6.2 References

PRC-PRO-AC-335, *Use and Control of Purchasing Card*

PRC-PRO-AC-40500, *Acquisition Management Plan*

PRC-PRO-EN-20050, *Engineering Configuration Management*

PRC-PRO-EN-40357, *Engineering Software Management*

PRC-PRO-IRM-10588, *Records Management Processes*

PRC-PRO-IRM-184, *Information Protection and Clearance*

PRC-PRO-IRM-592, *Unclassified Computer Security*

PRC-PRO-IRM-9679, *Administrative and Technical (Non-engineering) Document Control*

PRC-PRO-QA-052, *Issues Management*

Appendix A - Hanford Information System Inventory (HISI) Software Grading Checklist

The HISI's SGC is used to determine the software grade level. The software grade level and software type determine how to manage the various CHPRC software applications. Management of the software inventory is required by contract. The SGC is completed by answering "yes" or "no" to each question presented, as applicable.

Questions S1, S2, and S3 identify safety software classifications as defined in CRD O 414.1D. Software classification is a necessary part of the software inventory.

#	Question Details	Grade
S1	Is the software Safety System Software? Software for a nuclear facility that performs a safety function as part of a system, structure, or component (SSC) and is cited in either (1) a DOE approved documented safety analysis or (2) an approved hazard analysis per DOE P 450.4, "Safety Management System Policy" dated 10-15-96, and the DEAR Clause.	N/A
S2	Is the software Safety and Hazard Analysis Software and Design Software? Software that is used to classify, design, or analyze nuclear facilities. This software is not part of an SSC but helps to ensure the proper accident or hazards analysis of nuclear facilities or an SSC that performs a safety function.	N/A
S3	Is the software Safety Management and Administrative Controls Software? Software that performs a hazard control function in support of nuclear facility or radiological safety management programs or Technical Safety Requirements or other software that performs a control function necessary to provide adequate protection from nuclear facility or radiological hazards. This software supports eliminating, limiting, or mitigating nuclear hazards to workers, the public, or the environment as addressed in 10 CFR 830, 10 CFR 835, and the DEAR ISMS clause.	N/A

Questions 1 through 19 assign the software grade level based on the first "yes" answer. Software grade levels A, B, and C are classified within CHPRC to be "safety software." **All** questions in the SGC must be addressed and answered appropriately.

#	Question Details	Grade
1	Could software failure compromise a limiting condition of operation?	A
2	Could software failure cause a reduction in the safety margin for a safety SSC that is cited in DOE approved documented safety analysis (DSA)?	A

#	Question Details	Grade
3	Could software failure cause a reduction in the safety margin for other systems such as toxic or chemical protection systems that are cited in either (a) a DOE approved documented safety analysis or (b) an approved hazard analysis per DOE P 450.1, “Environment, Safety, and Health Policy for the Department of Energy Complex” and the DEAR ISMS clause?	A
4	Could software failure result in non-conservative safety analysis, design, or misclassification of facilities or SSCs?	A
5	Is the software a safety management database used to aid in decision making whose failure could impact safety SSC operation?	B
6	Could software failure result in incorrect analysis, design, monitoring, alarming, or recording of hazardous exposures to workers or public?	B
7	Could software failure compromise the defense in depth capability of a nuclear facility?	B
8	Could software failure cause a potential violation of regulatory permitting requirements?	C
9	Could software failure affect environment, safety, health monitoring or alarming systems?	C
10	Could software failure affect the safe operation of a SSC?	C
11	Is this software used to support facility protection (i.e., security, fire, etc.) or determine, display, or implement emergency actions?	C
12	Does the software perform nuclear material, hazardous chemical or waste inventory tracking and/or accountability?	C
13	Is the software used in support of a quality assurance program that implements a contractual requirement?	D
14	Is the software used to support a legal, regulatory, or external milestone?	D
15	Is the software used to engineer, analyze, or calculate general service facility equipment designs and/or configurations?	D
16	Is this software used to determine, select, or evaluate remedial actions for cleanup of contaminated sites or facilities?	D
17	Would a disruption of service, or software error or failure, result in costs greater than $100K to resolve?	D
18	Is software lifecycle cost through the first year of maintenance greater than $250K?	E

#	Question Details	Grade
19	Does the software/system contain information that may be considered “controlled use”?	E

If question 19 is answered yes, a controlled use exemption must be also identified

#	Question Details	Grade
E3	Exemption 3 - Statutory Exemption Definition: Exemption 3 is for information that is explicitly prohibited from disclosure, generally by federal law. Basing an OUO determination on Exemption 3 is very complex. Therefore, use of this exemption for Export Controlled Information (ECI) must be reviewed by an ECI Reviewer; all other Exemption 3 information should be reviewed by Legal Services.	N/A
E4	Exemption 4 - Commercial/Proprietary Definition: Exemption 4 addresses trade secrets and commercial or financial information that the Federal government has obtained from persons or commercial entities and that is privileged or confidential. This exemption is intended to protect both the interests of the government and persons submitting information to the government.	N/A
E5	Exemption 5 - Privileged Information Definition: Federal Government-generated information that would not be available by law other than through litigation, including letters, evaluations, plans, trade secrets, etc. are covered by Exemption 5. This exemption protects "intra-agency" and "inter-agency" communications, which has been interpreted by courts to include contractor privileged communications prepared under government direction or initiative (e.g., government funded).	N/A
E6	Exemption 6 - Personal Privacy Definition: Exemption 6 protects personal information related to a specific individual that, if disclosed, might cause personal distress or embarrassment.	N/A
E7	Exemption 7 - Law Enforcement Definition: Exemption 7 protects information compiled by an agency with the authority to enforce the law. It covers information compiled for law enforcement purposes regardless of the format of the information or how and where the information may be filed.	N/A

If the software application is considered Key Critical (KC) or Key Essential (KE) it must be identified on the checklist to ensure IRM Service Provider support.

#	Question Details	Grade
KC	Is this application, data storage location (share area) or telecommunication service necessary to manage, monitor and control during emergency and response situations including natural disasters, onsite unsafe conditions, re-establishing the safe work environment, avoiding imminent violation of safety and environmental requirements or restore security systems when an alternate is not available? Requires immediate restore within 24 hours.	N/A
KE	Is this application, data storage location (share area) or telecommunication service required for ensuring area, facility, business or plant safety and those actions necessary for continuing productive operation supporting the Hanford Site mission? Requires attention and restore within 48-72 hours.	N/A

If the software application is considered Safety Software a facility must be identified.

#	Question Details	Grade
NF1	Identify the specific nuclear facility where the application is used?	N/A

Appendix B - Software Grading Criteria

Software Quality Assurance (SQA) Work Activities

Software life cycle activities are closely integrated with SQA work activities, as described in CRD O 200.1A, *Information Technology Management*, and CRD O 414.1D. The required lifecycle products and SQA work activities are implemented based on grading levels and applicable software type.

Software owners are responsible for maintaining a project file of documentation supporting the performance of SQA work activities throughout the software lifecycle. The preferred method of maintaining software records is through the use of the IDMS. References to documentation produced during the software lifecycle are entered on the HISI Version Description tab. Approval for use of a software version is demonstrated through approved documentation and approvals on the HISI SGC and Version Description tabs. When updates are made to the HISI Version Description tab, an electronic record is written to the IDMS capturing the baseline metadata and document links for the particular version implemented.

The following table lists the SQA documentation items and the work activities typically covered in each document.

Version Description Document/Product	Typical Work Activities
Functional Requirements Document	• Software Project Management & Quality Planning
Alternatives Analysis	• Software Project Management & Quality Planning
Software Management Plan (Including Risk Management)	• Software Project Management & Quality Planning • Software Risk Management
Software Configuration Management Plan	• Software Configuration Management
Software Training Plan	• Training of Personnel in the Design, Development, Use and Evaluation of Safety Software
Acquisition Documents	• Procurement and Supplier Management
Software Requirements Specification	• Software Risk Management • Software Requirements Identification and Management • Verification and Validation
Software Design Description (including Software Safety)	• Software Risk Management • Software Design and Implementation • Software Safety
Requirements Traceability Matrix	• Software Requirements Identification and Management • Verification and Validation
Code Walkthrough	• Software Design and Implementation • Verification and Validation
User Documents	• Training of Personnel in the Design, Development, Use and Evaluation of Safety Software
Unit Testing	• Software Design and Implementation • Verification and Validation
Test Plan and Cases	• Verification and Validation
Acceptance Test Report	• Verification and Validation

Version Description Document/Product	Typical Work Activities
Contingency Plan	• Software Project Management & Quality Planning • Software Risk Management
Software Installation Plan	• Software Project Management & Quality Planning • Software Design and Implementation
User Qualification	• Training of Personnel in the Design, Development, Use and Evaluation of Safety Software
User Training	• Training of Personnel in the Design, Development, Use and Evaluation of Safety Software
Operational Testing	• Verification and Validation • Problem Reporting and Corrective Action
Change Request / Problem Report	• Problem Reporting and Corrective Action • Software Configuration Management
Retirement Plan/Checklist	• Software Project Management & Quality Planning • Software Configuration Management
Other Documents	• As applicable based on document topic.

Software Grading Levels

Software grade levels are assigned through the completion of the HISI SGC (refer to Appendix A).

Level A – Software applications that meet one or more of the following criteria.

1. Software failure that could compromise a limiting condition for operation.

2. Software failure that could cause a reduction in the safety margin for a safety SSC that is cited in DOE approved documented safety analysis.

3. Software failure that could cause a reduction in the safety margin for other systems such as toxic or chemical protection systems that are cited in either (a) a DOE approved documented safety analysis or (b) an approved hazard analysis per DOE P 450.1 and the DEAR ISMS clause.

4. Software failure that could result in non-conservative safety analysis, design, or misclassification of facilities or SSCs.

Level B – Software applications that do not meet Level A criteria but meet one or more of the following criteria.

1. Safety management databases used to aid in decision making whose failure could impact safety SSC operation.

2. Software failure that could result in incorrect analysis, design, monitoring, alarming, or recording of hazardous exposures to workers or the public.

3. Software failure that could comprise the defense in depth capability for the nuclear facility.

Level C – Software applications that do not meet Level B criteria but meet one or more of the following criteria.

1. Software failure that could cause a potential violation of regulatory permitting requirements.

2. Software failure that could affect environment, safety, health monitoring or alarming systems.

3. Software failure that could affect the safe operation of an SSC.

Level D – Software used to support quality affecting, general service applications, or business practice where software failure or application error could result in the potential for high business risk.

Level E – Software used in applications where software failure or application error could result in the potential for moderate project risk.

Graded Approach to SQA Documents/Products

Minimum SQA activities and documentation requirements are determined by the software grade level and software type. Based on the grade level and the software type, software owners are responsible for applying the appropriate degree of SQA work activities. In some cases, a software owner may determine that a higher level of SQA activity and documentation should be met beyond that which is required by the software grade and software type. Some factors to consider in determining a more rigorous level of SQA activity are: relationship to safety functions, software project size, lifecycle stage, complexity, cost, risk, visibility, and business and external impacts of the solution.

"**Full**" implementation of a particular work activity requires that the software owner complete all essential documentation to the degree necessary to ensure that the SQA work and lifecycle activity is performed in a traceable, planned, and orderly manner. The intended method for demonstrating full implementation of the work activities must be documented in the SMP and/or HISI Version Description Document tab, as applicable.

"**Graded**" implementation allows the software owner to use his/her best judgment in applying a graded approach to SQA work activities and documentation; however, it is important to note that each "graded" SQA documentation requirement must still be addressed. The graded approach can be accomplished by combining documents, preparing less formal documentation, and/or providing evidence of the work activity in another manner, such as email or other related documents. The intended graded implementation of the work activities must be documented in the SMP and/or HISI Version Description tab, as applicable.

In some cases work activities are not required, as indicated with "N/A." When work activities are performed by a service supplier, control of SQA activities of the software are achieved through procurement agreements and specification.

The graded approach of SQA documents/products, based on software types and grades, is set forth as indicated in the tables below:

1. Custom Developed Software

SQA Document/Product	Custom Developed Software Grade Level				
	A	B	C	D	E
Functional Requirements Document	Full	Full	Graded	Graded	Graded
Alternatives Analysis	Full	Full	Graded	Graded	
Software Management Plan	Full	Full	Graded	Graded	
Software Configuration Management Plan	Full	Full	Graded	Graded	
Acquisition Documents	Full	Full	Full	Graded	
Software Requirements Specification	Full	Full	Full	Graded	
Software Design Description	Full	Full	Full	Graded	Graded
Requirements Traceability Matrix	Full	Full	Full	Graded	
Code Walkthrough	Full	Full	Full	Graded	
User Documents	Full	Graded	Graded	Graded	
Unit Testing	Full	Full	Full	Graded	
Test Plan and Cases	Full	Graded	Graded	Graded	
Acceptance Test Report	Full	Graded	Graded	Graded	Graded
Contingency Plan	Full	Full	Graded	Graded	
Software Installation Plan	Full	Full	Full	Graded	
User Qualification	Full	Graded	Graded	Graded	
User Training	Full	Graded	Graded	Graded	
Operational (Periodic) Testing	Full	Graded	Graded	Graded	
Change Request / Problem Report	Full	Full	Full	Graded	
Retirement Plan/Checklist	Full	Full	Graded	Graded	Graded
Other Documents: • Data Security Plan[1]	Full	Graded	Graded	Graded	Graded
• Disaster Recovery Plan[2]	Full	Graded	Graded	Graded	Graded

[1] Required for software applications with databases, data, or output containing Controlled Use Information.
[2] Required for large-scale software applications.

2. Configurable Software

SQA Document/Product	Configurable Software Grade Level				
	A	B	C	D	E
Functional Requirements Document	Full	Full	Graded	Graded	Graded
Alternatives Analysis	Full	Full	Graded	Graded	
Software Management Plan	Full	Full	Graded	Graded	
Software Configuration Management Plan	Graded	Graded	Graded	Graded	
Acquisition Documents	Full	Full	Full	Graded	
Software Requirements Specification	Full	Full	Full	Graded	
Software Design Description	Full	Graded	Graded	Graded	
Requirements Traceability Matrix	Full	Full	Full	Graded	
Code Walkthrough	Full	Graded	Graded	Graded	
User Documents	Full	Graded	Graded	Graded	
Unit Testing	Full	Graded	Graded	Graded	
Test Plan and Cases	Full	Graded	Graded	Graded	
Acceptance Test Report	Full	Graded	Graded	Graded	Graded
Contingency Plan	Full	Full	Graded	Graded	
Software Installation Plan	Full	Full	Graded	Graded	
User Qualification	Full	Graded	Graded	Graded	
User Training	Full	Graded	Graded	Graded	
Operational (Periodic) Testing	Full	Graded	Graded	Graded	
Change Request / Problem Report	Full	Full	Graded	Graded	
Retirement Plan/Checklist	Full	Full	Graded	Graded	Graded
Other Documents: • Data Security Plan[1]	Full	Graded	Graded	Graded	Graded
• Disaster Recovery Plan[2]	Full	Graded	Graded	Graded	Graded

[1] Required for software applications with databases, data, or output containing Controlled Use Information.

[2] Required for large-scale software applications.

3. Acquired Software

SQA Document/Product	Acquired Software Grade Level				
	A	B	C	D	E
Functional Requirements Document	Graded	Graded	Graded	Graded	Graded
Alternatives Analysis	Graded	Graded	Graded	Graded	
Software Management Plan	Full	Graded	Graded	Graded	
Software Configuration Management Plan	Graded	Graded	Graded	Graded	
Acquisition Documents	Full	Full	Full	Graded	
Software Requirements Specification	Full	Full	Full	Graded	
Software Design Description	N/A	N/A	N/A	N/A	
Requirements Traceability Matrix	Full	Full	Full	Graded	
Code Walkthrough	N/A	N/A	N/A	N/A	
User Documents	Full	Graded	Graded	Graded	
Unit Testing	N/A	N/A	N/A	N/A	
Test Plan and Cases	Full	Graded	Graded	Graded	
Acceptance Test Report	Full	Graded	Graded	Graded	Graded
Contingency Plan	Full	Graded	Graded	Graded	
Software Installation Plan	Graded	Graded	Graded	Graded	
User Qualification	Full	Graded	Graded	Graded	
User Training	Full	Graded	Graded	Graded	
Operational (Periodic) Testing	Full	Graded	Graded	Graded	
Change Request / Problem Report	Full	Full	Graded	Graded	
Retirement Plan/Checklist	Graded	Graded	Graded	Graded	Graded
Other Documents: • Data Security Plan[1]	Full	Graded	Graded	Graded	Graded
• Disaster Recovery Plan[2]	Full	Graded	Graded	Graded	Graded

[1] Required for software applications with databases, data, or output containing Controlled Use Information.

[2] Required for large-scale software applications.

4. Utility Calculation Software

SQA Document/Product	Utility Calculation Software Grade Level				
	A	B	C	D	E
Functional Requirements Document	Graded	Graded	Graded	Graded	Graded
Alternatives Analysis	Graded	Graded	Graded	Graded	
Software Management Plan	Full	Graded	Graded	Graded	
Software Configuration Management Plan	Graded	Graded	Graded	Graded	
Acquisition Documents	Full	Full	Full	Graded	
Software Requirements Specification	Full	Full	Full	Graded	
Software Design Description	Full	Graded	Graded	Graded	
Requirements Traceability Matrix	Full	Full	Full	Graded	
Code Walkthrough	Graded	Graded	Graded	Graded	
User Documents	Full	Graded	Graded	Graded	
Unit Testing	Graded	Graded	Graded	Graded	
Test Plan and Cases	Graded	Graded	Graded	Graded	
Acceptance Test Report	Graded	Graded	Graded	Graded	Graded
Contingency Plan	Full	Graded	Graded	Graded	
Software Installation Plan	Graded	Graded	Graded	Graded	
User Qualification	Full	Graded	Graded	Graded	
User Training	Full	Graded	Graded	Graded	
Operational (Periodic) Testing	Graded	Graded	Graded	Graded	
Change Request / Problem Report	Graded	Graded	Graded	Graded	
Retirement Plan/Checklist	Graded	Graded	Graded	Graded	Graded
Other Documents: • Data Security Plan[1]	Full	Graded	Graded	Graded	Graded
• Disaster Recovery Plan[2]	Full	Graded	Graded	Graded	Graded

[1] Required for software applications with databases, data, or output containing Controlled Use Information.

[2] Required for large-scale software applications.

5. Commercial Design & Analysis Software

SQA Document/Product	Commercial Design & Analysis Software Grade Level				
	A	B	C	D	E
Functional Requirements Document	N/A	N/A	N/A	N/A	N/A
Alternatives Analysis	N/A	N/A	N/A	N/A	
Software Management Plan[1]	N/A	N/A	N/A	N/A	
Software Configuration Management Plan	Graded	Graded	Graded	Graded	
Acquisition Documents	Full	Full	Full	Graded	
Software Requirements Specification	Full	Full	Full	Graded	
Software Design Description	N/A	N/A	N/A	N/A	
Requirements Traceability Matrix	N/A	N/A	N/A	N/A	
Code Walkthrough	N/A	N/A	N/A	N/A	
User Documents	N/A	N/A	N/A	N/A	
Unit Testing	N/A	N/A	N/A	N/A	
Test Plan and Cases	N/A	N/A	N/A	N/A	
Acceptance Test Report	N/A	N/A	N/A	N/A	N/A
Contingency Plan	N/A	N/A	N/A	N/A	
Software Installation Plan	N/A	N/A	N/A	N/A	
User Qualification	N/A	N/A	N/A	N/A	
User Training	N/A	N/A	N/A	N/A	
Operational (Periodic) Testing	N/A	N/A	N/A	N/A	
Change Request / Problem Report	Full	Full	Graded	Graded	
Retirement Plan/Checklist	Graded	Graded	Graded	Graded	N/A
Other Documents: • Data Security Plan[2]	Full	Graded	Graded	Graded	Graded
• Disaster Recovery Plan[3]	Full	Graded	Graded	Graded	Graded

[1] Software owners of Commercial Design & Analysis software should document intended application of graded approach in the HISI Version Description tab.

[2] Required for software applications with databases, data, or output containing Controlled Use Information.

[3] Required for large-scale software applications.

Appendix C - Glossary

Term	Definition
Acquired Software	Acquired software is generally supplied through basic procurements, two-party agreements, or other contractual arrangements. Acquired software includes commercial off-the-shelf (COTS) software, such as operating systems, database management systems, compilers, software development tools, and commercial calculation software and spreadsheet tools, as well as DOE "toolbox" codes. Downloadable software that is available at no cost to the user (referred to as freeware) is also considered acquired software. Firmware is acquired software and is generally provided by a hardware supplier through the procurement process. In most cases, firmware cannot be directly modified by the user after receipt from the vendor.
Acceptance Test	(1) The process of exercising or evaluating a system or system component by manual or automated means to ensure that it satisfies the specified requirements and to identify differences between expected and actual results in the operating environment. (2) Formal testing conducted to determine whether or not a software product or system satisfies its acceptance criteria and to enable the system owner to determine whether or not to accept the product or system.
Application	Project/Facility/Function system or computer-based computation for which software is designed to fulfill specific needs of the user; for example, software for measurement, design, research & development, business administration, or process control (e.g., RadCalc, AutoCad, Algor, PassPort, LabView).
Baseline	A set of configuration items (software components and documents) that has been formally reviewed and approved for use, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.
Change Control	An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification.
Code	Computer instructions and data definitions expressed in a programming language or in a form that is output by an assembler, compiler, or other translator. A combination of programming language constructs and natural language used to express a computer program design. Information which changes the output or processing of a software application (e.g., user-defined parameters, scripts, schemas, etc.).

Term	Definition
Commercial Design and Analysis Software	Commercial design and analysis software is used in conjunction with design and analysis services provided by a commercial contractor. The design service provider uses it's independently developed or acquired software and delivers a completed design. Procurement contracts can be enhanced to require that the software used in the design or analysis services meet the requirements.
Commercial Off-the-Shelf (COTS) Software	COTS software is purchased software that is typically used "as is." COTS products are designed to be easily installed and to interoperate with existing system components.
Configuration Item	An aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration management process.
Configurable Software	Configurable software is commercially available software or firmware that allows the user to modify the structure and functioning of the software in a limited way to suit user needs. An example is software associated with programmable logic controllers (PLCs).
Configuration Management	(1) The process of identifying and defining the configuration items in a system (i.e., software and hardware), controlling the release and change of these items throughout the system's life cycle, and recording and reporting the status of configuration items and change requests. (2) A discipline that effectively controls and manages all modifications to a software component, product, or system. Technical and administrative processes and tools are used to identify and document the functional and physical characteristics of the configuration items, manage and track changes to those items, record and report change processing and implementation status, and verify compliance with specified requirements.
Custom Developed Software	Custom developed software is built specifically for an application or to support a specific function. Custom applications may be developed by DOE, one of its contractors, or contracted with a qualified software company through the procurement process. Examples of custom developed software includes material inventory and tracking database applications, accident consequence applications, control system applications, and embedded custom developed software that controls a hardware device.

Term	Definition
Error	**(1)** The difference between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. An incorrect result. **(2)** A condition deviating from an established baseline, including deviations from the current approved computer program and its baseline requirements.
Flat-File	A spreadsheet workbook file that contains a single worksheet and does not contain any external references (e.g., linked or embedded objects) for data and functions required to perform the required calculations.
Full	Implementation of a particular work activity that requires the software owner to complete all essential documentation to the degree necessary to ensure that the SQA work and lifecycle activity is performed in a traceable, planned, and orderly manner. The intended method for demonstrating full implementation of the work activities must be documented in the Software Management Plan (SMP) and/or HISI Version Description Document tab, as applicable.
Functional Requirement	A defined objective or characteristic action that a system or component of the system must be able to perform. For example, a system may have inventory control as its primary functional requirement.
Graded	Implementation of a particular work activity that allows the software owner to use his/her best judgment in applying a graded approach to the SQA work activities and documentation. Each “graded” SQA documentation requirement must be addressed. The graded approach can be accomplished by combining documents, preparing less formal documentation, and/or providing evidence of the work activity in another manner, such as email or other related documents. The intended graded implementation of the work activities must be documented in the SMP and/or HISI Version Description tab, as applicable.
Independent Technical Reviewer	A competent individual who did not work on the original software development or modification but who may be from the same organization. The Independent Technical Review may be performed by the originator’s supervisor, provided the supervisor did not specify a singular design approach or rule out certain design considerations and did not establish the design inputs used on the software development, provided the supervisor is the only individual in the organization competent to perform the verification.

Term	Definition
Large-scale Installation	A software application is considered to be a large-scale installation if the modification, acquisition, or lifecycle cost of the software will exceed $250,000.
Legacy Software	Software used to perform analysis, calculations, reporting, or operation of hardware or a facility which was not originally (1) developed in accordance with a recognized national consensus software QA standard, or (2) developed when the history or pedigree of the application is unknown, or (3) approved for use under the latest version of this procedure.
Parameter	A variable that is given a constant value for a specified application. A constant, variable or expression used to pass values between software modules. A parameter may be an internal setting or set point in a computer system that is set by the user within the operational range provided by the vendor software.
Regression Testing	Selective retesting to detect errors introduced during modification of the computer program or to verify that the modified computer program still meets its specified requirements.
Requirements Traceability Matrix	A RTM is a representation of software requirements information that provides a many-to-many map from and to each requirement, and to and from various system and software artifacts. The purpose of the RTM is to trace software requirements through definition, design, and testing throughout the lifecycle of the software.
Safety Software	Includes safety analysis and design software, safety system software, and safety management and administrative controls software. (1) Safety analysis and design software is used for the design and analysis of safety structures, systems, and components for the facility. These applications include software used in structural; electrical; mechanical; heating; ventilation and air conditioning; criticality safety; fire protection design and analysis and analysis decisions. (2) Safety system software is used for I&C, where software and firmware provide control and monitoring functionality for such components as valves and switches, including all PLC's, SCADA's, DCS and hybrid systems consisting of a mix of DCS and PLC/Human Machine Interface (HMI) features. (3) Safety management and administrative controls software performs a hazard control function in support of nuclear facility safety or radiological safety programs. It supports eliminating, limiting, or mitigating nuclear hazards to workers, the public, and the environment.

Term	Definition
Software	Computer programs, and code, including databases, spreadsheets, macros, routines, and web pages (both static and dynamic) developed to fulfill specific user needs, such as data maintenance, data manipulation, calculations, and reporting. Software is supported by documentation that describes its functions, uses, and maintenance (such as plans, requirements documentation, user instructions, and data dictionaries).
Software Lifecycle	The period of time that starts when a software product is conceived and ends when the software product is no longer available for routine use. The software lifecycle typically includes a concept phase, a requirements phase, an evaluation/design phase, an implementation phase, a test phase, an installation phase, an operations and maintenance phase, and a retirement phase.
Software Life Cycle Methodology	A set of guidelines for managing, developing, enhancing, or maintaining computer-based systems, including the steps, techniques, deliverables, checkpoints, and roles and responsibilities.
Software Management Plan	A plan for the development of software products necessary to provide confidence that the software conforms to established requirements.
Software Owner	The responsible individual designated as the point of contact for computer software development, maintenance and utilization.
Software SME	The responsible individual designated by the Project, Function, or Facility Manager as the point of contact for the project/function/facility software management.
Software Validation (Test)	The process of testing and evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. Validation involves running test cases to ensure the computed output meets specified expectations and requirements, including numerical correctness of the results on the bases of comparison with alternate calculations (e.g., hand calculations, analytical solutions, and other computer code). Validation testing may include, but is not limited to, system integration, acceptance testing and operational testing.

Term	Definition
Software Verification (Review)	Verification is the process of determining if the products of a given life cycle phase satisfy the conditions imposed at the start of the phase. It is the review necessary to verify the adequacy of documentation. Verification involves the comparison of computer software and application components, including associated documentation, with functional requirements and specifications for these components, to ensure the planned equations and logic are incorporated into the application, that the documentation includes the required topics and sections, and that the application is useful and accurate.
System Software	Software designed to enable the operations and maintenance of a computer system and its associated computer programs.
Testing	The process of; (a) operating a system (i.e., software and hardware) or system component under specified conditions, (b) observing and recording the results, and (c) making an evaluation of some aspect of the system (i.e., software and hardware) or system component in order to verify that it satisfies specified requirements and to identify errors
Test Case	A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.
Test Plan	A document that describes the approach to be followed for testing a system or component. Typical contents identify the items to be tested, tasks to be performed, and responsibilities for the testing activities.
Toolbox Code	"Toolbox" code is software that is typically developed by the technical staff of the government agency for which it is created, or is developed by an external entity, with funding and specifications from the agency. Because agencies can directly control all aspects of "toolbox" code products, these are generally preferred for government purposes. DOE toolbox codes have widespread use and are of appropriate qualification for use within DOE. These codes are verified and validated and constitute a "safe harbor" methodology. A list of toolbox codes comprising the DOE Safety Software Central Registry can be accessed at: http://energy.gov/hss/information-center/department-energy-technical-standards-program/safety-software-quality.

Term	Definition
Utility Calculation Software	Utility calculation software typically uses COTS spreadsheet applications as a foundation and user developed algorithms or data structures to create simple software products. The utility calculation software within the scope of this document is used frequently to perform calculations associated with the design of a SSC. Utility software that is used with high frequency may be labeled as custom software and may justify the same safety SQA work activities as custom developed software. With utility calculation software, it is important to recognize the difference between QA of the algorithms, macros, and logic that perform the calculations versus QA of the COTS software itself. Utility calculation software includes the associated data sets, configuration information, and test cases for validation and/or calibration.