Manual Number: 1Q

Manual Title: *Quality Assurance Manual*

Revision Summary

1.	**Document and Revision Numbers:** Procedure 20-1, Rev. 11
2.	**Document Title:** Software Quality Assurance
3.	**Effective Date:** 12/17/08
4.	**Document Changes (Editorial):** Global: Revised title of MRP 4.23 to "Corrective Action Program". **References** Deleted "revision number" on "Department of Energy (DOE)/NNSA QC-1, Weapon Quality Policy QC-1" and "Department of Energy (DOE) RW DOE/RW-0333P, 'Quality Assurance Requirements and Description for the Civilian Radioactive Waste Management Program'"
5.	**Training Requirements:** As with any procedure revision, those employees affected by the procedure need to familiarize themselves with the changes. No additional training is required.

Software Quality Assurance

APPROVED
COMPANY-LEVEL
PROCEDURE

Purpose

This procedure defines the requirements and responsibilities for a standard systematic approach for the quality control of computer software.

This procedure also establishes a mechanism for the Performing Entity to meet the applicable requirements in support of contractual obligations. For a current list of Source Document references, go to the Standards/Requirements Identification Document (S/RID) webpage accessible through ShRINE.

Scope

The provisions of this procedure apply to members of the Performing Entity for management and operations at the Savannah River Site (SRS), and to subcontractors performing work for any member of the Performing Entity when required by subcontract or applicable law that develop, procure, maintain, operate, use, or retire software. It defines the extent and level of software life cycle controls and software design verification activities related to software based on the software classification of the software application. The software classification is based on the intended use of the software.

This procedure applies to all software and firmware except:

- Software which is part of measuring and test equipment (M&TE) or measuring systems and equipment (MS&E) where the equipment is verified periodically as part of a calibration program compliant with Procedure Manual 1Q, *Quality Assurance Manual*, Procedure 12-1, "Control of Measuring and Test Equipment" or Procedure Manual 1Q, Procedure 2-7, "QA Program Requirements for Analytical Measurement Systems" respectively
- Engineering Calculations under control of Procedure Manual E7, *Conduct of Engineering and Technical Support, Procedure* 2.31, "Engineering Calculations"
- Software used exclusively for scoping activities as defined by Procedure Manual 1Q, Procedure 2-3, "Control of Research and Development Activities"
- Firmware that personnel can not change or control i.e. embedded in the system hardware. This firmware is covered under the QA requirements for that system. This firmware is tested and validated as part of the system it is embedded within.

Terms and Definitions

See Procedure Manual 1Q, Appendix A, "Glossary of Terms" for Terms and Definitions

Responsibilities

Managers

Managers are responsible for:

- ensuring a software quality assurance program is established, documented, and implemented in accordance with this procedure,
- designating individuals or organizations responsible for implementing this procedure and define the interfaces with external organizations.

Cognizant Technical Functions (CTFs)/Design Agency

CTFs/Design Agency are responsible for:

- ensuring software classification is determined and documented for the intended use of software (OSR 19-337),
- ensuring Software Quality Assurance (QA) procedures/plans are reviewed and approved by CTF/Design Authority Management and the Software Owner,
- ensuring the safety software requirements (Attachment D) are reviewed for the Safety Software Inventory List (SSIL).

Cognizant Quality Functions (CQFs)

CQFs are responsible for:

- reviewing and approving Software QA procedures/plans,
- reviewing and approving the software classification document (OSR 19-337),
- providing approved software classification document identifying software as Safety Software to the Performing Entity QA manager for inclusion in the SSIL.

Software Owner/Design Authority

Software Owner/Design Authority is responsible for:

- assigning software classification,
- reviewing and approving Software QA procedures/plans,
- reviewing and approving the software classification document (OSR 19-337).

Performing Entity Quality Manager

Performing Entity Quality Manager is responsible for:

- the Software Quality Assurance program,
- generating and maintaining the Safety Software Inventory List,
- ensuring the Safety Software Inventory List is reviewed and updated annually.

Procedure

This procedure comprises the following Sections:

A. General
B. Software Classification Requirements
C. Software Quality Assurance Procedures/Plans
D. Software Life Cycle
E. Software Configuration Control
F. Evaluation (Existing/Acquired Software)
G. Software Procurement
H. Problem Reporting and Corrective Action
I. Software Security Controls
J. Safety Software Inventory List (SSIL)

A. General

1. Management ensures that a software quality assurance program is established, documented, and implemented. They designate individuals or organizations responsible for implementation and define

the interfaces with external organizations. Management ensures that these individuals receive training that is commensurate with the scope, complexity, and importance of the task.

2. Software shall be controlled throughout its life cycle, using a graded approach, based on its software classification. It is not the intent of this procedure to restrict the software life cycle methodology defined in implementing procedures, provided that the chosen methodology encompasses the activities, documentation, reviews, and approvals required by this procedure.

3. The graded approach to Software Quality Assurance is based on a classification scheme used for managing the software through out the life cycle based on its intended function. The two types of classification are:

   - software designated as part of a Structure, System, or Component (SSC) with an assigned functional classification, or
   - software that is not designated as part of a SSC, using A through E classification.

B. Software Classification Requirements

1. The CTF shall determine which classification process applies (SSC Functional Classification or A through E Classification).

2. Software that is part of an SSC which has a Component Location Identifier (CLI) Number (as defined in Procedure Manual E7, Procedure 1.30 "Component Numbering System") is classified using the Functional Classification Procedure Manual E7, Procedure 2.25, "Functional Classifications." The Life Cycle Documentation associated with these software classifications is provided in Attachment B, "Life Cycle Documentation Requirements Matrix/SQA Graded Approach."

3. Classification of software using the A through E designation is detailed in Attachment A, "Graded Approach to Software Classification." The Life Cycle Documentation associated with these software classifications is provided in Attachment B, "Life Cycle Documentation Requirements Matrix/SQA Graded Approach."

4. Detailed implementation processes based on software classification are provided in specific manuals (for example, E7, 12B) and/or individual software implementation plans.

5. The CTF will ensure that the software quality assurance implementation process meets the requirements throughout the entire software life cycle.

6. The Software Owner/Design Authority assigns the classification. The results of the classification process are documented on the Software Classification Document (OSR 19-337) by the CTF/Design Agency and approved by the CQF. Attachment D is used to determine if the software is required to be on the SSIL (see Attachment C).

7. The CQF will verify that the software classification has been established and approves the software classification. The CQF will provide the approved software classification document identifying software as Safety Software to the Performing Entity QA manager for inclusion in the SSIL (Attachment C).

C. Software Quality Assurance Procedures/Plans

1. Based on the nature, complexity, and intended use of the software, the CTF shall ensure that procedures/plans for software quality assurance are prepared which identify the following:

   a. the software products to which it applies

   b. the organizations responsible for performing the work and achieving software quality, and their tasks and responsibilities

   c. the software engineering methods

d. required documentation

e. standards, conventions, techniques, or methodologies which shall be used to guide the software development, as well as methods to assure compliance to the same

f. the required software reviews

g. the methods for error reporting and corrective action

2. These procedures/plans may be prepared individually for each software project, or may exist as a generic document to be applied to software prepared within, procured, or used by each organization. Software QA procedures/plans shall be reviewed and approved by CTF/Design Authority management and the CQF. Optionally, these requirements may be distributed within the software life cycle documentation specified in Section D. If such requirements are distributed in life cycle documentation, the CQF shall review and approve these documents to ensure that such requirements are addressed prior to the implementation phase.

D. Software Life Cycle

1. Introduction

   Software shall be controlled in a traceable, planned, and orderly manner. The software life cycle defined in this section provides the basis for planning and implementing a software development, maintenance, or application project. This section identifies the specific software activities, documentation, and reviews associated with each life cycle phase. The identified life cycle phase requirements shall be applied using the graded approach appropriate for the software. However, no strict chronological constraints exist between the requirements identified. The number of phases and relative emphasis placed on each phase of software development or maintenance shall be defined in the software QA procedure/plan and will be dependent on the nature and complexity of the software. As each life cycle document is approved, it shall be placed under configuration control in accordance with Section E.

2. Requirements Phase

   a. During this phase the requirements that the software must satisfy shall be specified, documented, reviewed, and approved. These requirements shall define the functions to be performed by the software and shall provide the detail and information necessary to design the software.

   b. Software requirements shall be verifiable and traceable throughout all stages of the software development cycle.

   c. Documentation

      Software requirements documentation shall define requirements for, functionality, performance, design inputs, design constraints, installation considerations, operating systems (if applicable), and external interfaces necessary to design the software. Acceptance criteria shall be established in the software requirements documentation for each of the identified requirements. Such criteria shall be used for verification/validation planning and performance as defined in each related life cycle phase.

   d. Review and Approval

      The CTF shall assure that the software requirement documentation is reviewed and approved by the responsible organization at the completion of this phase. This review shall assure that the requirements are complete, verifiable, consistent, and technically feasible. The review shall also assure that the requirements will result in a feasible and usable final product.

3. Design Phase

   a. During this phase a software design shall be developed, documented, reviewed, and controlled. The responsible design organization shall prescribe and document the design activities to the level of detail necessary to permit the design process to be carried out and to permit verification that the design meets requirements.

   b. Design Elements

      (1) The design shall specify the interfaces, overall structure (control and data flow) and the reduction of the overall structure into physical solutions (algorithms, equations, control logic, and data structures).

      (2) Computer programs are designed as an integral part of an overall system. Therefore, software design shall consider the computer program's operating environment.

      (3) Measures to mitigate the consequences of problems shall be an integral part of software design. These potential problems include external and internal abnormal conditions and events that can affect the computer program

   c. Documentation

      Software design documentation shall contain:

      (1) a description of the major components of the software design as they relate to the software requirements;

      (2) a technical description of the software with respect to the theoretical basis, mathematical model, control flow, data flow, control logic, data structure, numerical methods, physical models, process flow, process structures, and applicable relationship between data structure and process standards;

      (3) a description of the allowable or prescribed ranges for inputs and outputs;

      (4) the design described in a manner that can be translated into code; and

      (5) a description of the approach to be taken for intended test activities based on the requirements and design that specify the hardware and software configuration to be used during test execution.

   d. Review and Approval

      (1) The organization responsible for the design shall identify and document particular verification methods to be used and assure that an Independent Review (IR) is performed and documented. This review shall evaluate the technical adequacy of the design approach; assure internal completeness, consistency, clarity, and correctness of the software design; and verify the software design is traceable to the requirements.

      (2) The organization responsible for the design shall also assure that the test results adequately demonstrate the requirements have been met (See Section D. 5. "Test Phase").

      (3) The IR shall be performed by competent individual(s) other than those who developed and documented the original design, but who may be from the same organization. The results of the IR shall be documented with the identification of the verifier indicated. When review alone is not adequate to determine if requirements are met, alternate calculations shall be used, or tests shall be developed and integrated into the appropriate activities of the software development cycle. Software design documentation shall be completed prior to finalizing the IR.

(4) The extent of the IR and the methods chosen are a function of:

(a) the importance to safety,

(b) the complexity of the software,

(c) the degree of standardization, and

(d) the similarity with previously proven software.

4. Implementation Phase

a. Implementation Phase

During this phase the implementation process shall result in software products such as computer program listings and instructions for computer program use. The implemented software shall be analyzed to identify and correct errors. The source code for development of software classified A/B or SC/SS shall be placed under configuration control in accordance with Section E (Software Configuration Control) prior to commencement of the Test Phase.

b. Documentation

Implementation documentation shall include a copy of the software, test cases and associated criteria that are traceable to the software requirements, and design documentation.

c. Review and Approval

The CTF shall assure that specified design constraints, standards, and conventions are implemented. In addition, the CTF shall assure that reviews of the test cases are performed and the test cases are approved by the responsible organization at the completion of this phase.

5. Test Phase

a. During this phase the software shall be validated by executing the test cases. Failure to successfully execute the test cases shall be reviewed to determine if modification of the requirements, the design, the implementation, or the test plans and test cases are required. Testing shall demonstrate the capability of the software to produce valid results for test cases encompassing the range of permitted usage defined by the program documentation. Such activities shall ensure that the software adequately and correctly performs all intended functions.

b. Testing shall demonstrate, as appropriate, that the computer program:

(1) properly handles abnormal conditions and events as well as credible failures

(2) does not perform adverse unintended functions; and

(3) does not degrade the system either by itself, or in combination with other functions or configuration items.

c. Test Phase activities shall consist of the testing of the software to assure adherence to requirements, and to assure that the software produces correct results for the test cases specified. Acceptable methods for evaluating the adequacy of the software test case results include:

(1) analysis without computer assistance

(2) other validated computer program(s),

(3) experiments and tests,

(4) standard problems with known solutions,

(5) confirmed published data and correlation’s

d. Documentation

Test Phase documentation shall include test procedures or plans and the results of the execution of test cases. The test results documentation shall demonstrate successful completion of all test cases or the resolution of unsuccessful test cases and provide direct traceability between the test results and specified software requirements.

e. Test procedures or plans shall specify the following, as applicable:

(1) required tests and test sequence,

(2) required range of input parameters,

(3) identification of the stages at which testing is required,

(4) requirements for testing logic branches,

(5) requirements for hardware integration,

(6) anticipated output values,

(7) acceptance criteria,

(8) reports, records, standard formatting, and conventions,

(9) identification of operating environment, support software, software tools or system software,

(10) Hardware Operating System(s) and/or limitation.

f. Review and Approval

The CTF shall ensure that a technical review of the test procedures/plans and test results are performed. The technical review of the test results shall ensure that the test requirements have been satisfied. When software design is required, Independent Technical Review (ITR) of the test results is required in accordance with the design phase section of this procedure.

6. Installation and Acceptance Phase

a. During this phase the software becomes part of a system incorporating applicable software components, hardware, and data and is accepted for use. For the installation and acceptance phase:

(1) The CTF shall determine the acceptance testing to be performed.

(2) Acceptance testing shall include a comprehensive test in the operating environment.

(3) Acceptance testing shall be performed prior to approval of the computer program for use.

(4) Software validation shall be performed to ensure that the installed software product satisfies the specified software requirements. The engineering function shall determine the acceptance testing to be performed prior to approval of the computer program for use.

b. Documentation

Installation and acceptance phase documentation shall include results of the execution of test cases for system installation and integration, user instructions, and documentation of the acceptance of the software for operational use.

c. User instructions shall include:

(1) approved operating systems

(2) a description of the user's interaction with the software,

(3) a description of any required training necessary to use the software,

(4) input and output specifications,

(5) input and output formats,

(6) a description of software and hardware limitations,

(7) a description of user messages initiated as a result of improper input and how the user can respond,

(8) information for obtaining user and maintenance support.

d. Review and Approval

The CTF shall assure that design verification activities and a technical review of the acceptance testing, installation results, and user instructions is performed and completed prior to computer program use. Installation testing shall ensure the integrity of the software and its interfaces (for example, associated system and memory resident software, associated run-time libraries, and the hardware configuration). If any of these dependent features are changed in any way, installation testing shall be re-performed. The documentation of the acceptance of the software for operational use shall ensure that configuration baselines, documentation, and reviews have been completed.

7. Operations and Maintenance Phase

a. During this phase, software shall be controlled to remove latent errors (corrective maintenance), to respond to new or revised requirements (preventive maintenance), or to adapt the software to changes in the operating environment (adaptive maintenance). Software modifications shall be approved, documented, verified and validated, and controlled in accordance with the related life cycle phases.

b. The validation of modifications shall be subject to selective regression testing to detect errors introduced during the modification of software or operating system components to verify that the modifications have not caused unintended adverse effects and to verify that the modified software still meets its specified requirements.

c. Test cases shall be developed and documented to permit confirmation of acceptable performance of the software in the environment in which the software is used. Test cases shall be run whenever the software is installed on a different computer, or when significant hardware or operating system configuration changes are made.

d. Periodic in-use manual or automatic self-check in-use tests shall be prescribed and performed for those computer programs where computer program errors, data errors, computer hardware failures, or instrument drift can affect required performance.

8. Retirement Phase

During the retirement phase, the support for a software product shall be terminated and the routine use of the software prevented.

E. Software Configuration Control

1. Introduction

The methods to be used to control, uniquely identify, describe, and document the configuration of each version or update of a computer program (for example, source, object, back-up files) and its related documentation (for example, software design requirements, instructions for computer program use, test plans, and results) shall be described in implementing procedures. Such

procedures shall meet the following criteria for configuration identification, change control and configuration status accounting.

2. Configuration Identification

   a. A configuration baseline shall be defined at the completion of each major phase of the software life cycle. Approved changes created subsequent to a baseline shall be added to the baseline. A baseline shall define the most recently approved software configuration.

   b. A baseline labeling system shall be implemented that:

      (1) uniquely identifies each configuration item,

      (2) identifies changes to configuration items by revision,

      (3) provides the ability to uniquely identify each configuration of the revised software available for use.

3. Configuration Change Control

   a. Proposed changes to software shall be formally documented. This documentation shall contain a description of the change, the rationale for the change, requirements for re-testing and acceptance of the test results, and the identification of other life cycle documentation that will require modification to address the proposed change. For example, a change in the design documentation may require a change in the requirements, implementation, and user documentation.

   b. The change shall be evaluated and approved for release by the organization responsible for the original design, unless an alternative organization has been given the authority to approve the changes. Modification to the life cycle documentation will require the same level of review and approval as the original. Only approved changes shall be made to software baselines. Software verification activities shall be performed for the change as necessary to ensure the change is appropriately reflected in software documentation, and to ensure that document traceability is maintained. Software validation shall be performed as necessary to ensure that the change does not adversely affect the performance of the software.

   c. Control of the baseline includes describing check-in/out of the source code. Software backups shall use a proven method to ensure recovery capability.

4. Configuration Status Accounting and Control

   The information that is needed to manage a configuration baseline shall be documented. This information shall identify the approved configuration baseline, the status of proposed changes to the configuration baseline, the status of approved changes, and information to support the functions of configuration identification and configuration control. Configuration control is to include configuration status notification of organizations affected by configuration changes.

5. Cyber Security Configuration Management

   The cyber security configuration management polices, programs and procedures, which include software security configuration, shall be followed as defined in Procedure Manual 10Q, Computer Security Manual.

F. Evaluation (Existing/Acquired Software)

1. Software, which was not developed in accordance with this procedure, shall be classified, evaluated, validated, placed under configuration control, and controlled in accordance with the life cycle requirements specified in the applicable attachment. Management shall ensure that the evaluation schedule for affected software is developed. The CTF shall perform and document an evaluation of existing software which:

a. determines the adequacy of software documentation to support testing, operation, and maintenance.

b. identifies activities to be performed throughout the applicable life cycle of the software including preparation of required documentation and performance of required reviews and/or tests,

c. determines the software's capabilities and limitations for intended use,

d. specifies test plans and test cases required to validate the capabilities within the stated limitations,

e. identifies instructions for software use within the limits of its capabilities,

f. identifies any exceptions to the life cycle documentation and its justification.

2. This evaluation may be documented in a work request, plan, procedure, project level instruction, or other method, as appropriate. Exceptions to the life cycle documentation shall be approved by the CQF.

3. As an alternative, the user organization shall obtain the above documentation from the supplier or perform a documented review of the documentation at the supplier facility to determine acceptability.

4. Revisions to previously baseline software received from organizations not required to follow Computer Software QA Requirements shall be evaluated in accordance with the requirements.

5. The results of the above documentation and the performance of the actions necessary to accept the software shall be reviewed and approved by CTF/Design Authority. The documentation and associated computer program(s) shall establish the current baseline.

G. Software Procurement

1. Procurement

a. Software and software services shall be procured in accordance with procurement procedures. Procurement documents shall identify requirements for Supplier's reporting of software errors to the Purchaser and, as appropriate, the Purchaser's reporting of software errors to the Supplier.

b. Procurement of software, including development of specifications, shall be in accordance with the requirements of Procedure Manual 3E, Procurement Specification Procedure Manual, and the following matrix.

A through E Classified Software	**SSC Classified Software**	**Minimum Procurement Level**
A	SC	1
B	SS	2
C	PS	3
D or E	GS	3

c. Software intended for commercial dedication may be procured as a standard commercial grade item. Procured software shall be classified, placed under configuration control, and controlled in accordance with the life cycle requirements prior to installation.

2. Dedication of Commercial Grade Software

   Organizations planning to use off-the-shelf software in applications classified as A/B or SC/SS shall review the intended application sufficiently to determine the critical functions that provide evidence of the software's suitability for use. Once the critical functions have been established, the user or support organization's CTF shall define the methods to verify their adequacy and provide verifiable acceptance criteria. Acceptable dedication methods and required dedication documentation shall be prepared in an appropriate manner similar to Procedure Manual 1Q, Procedure 7-3, "Commercial Grade Item Dedication.

H. Problem Reporting and Corrective Action

1. The problem reporting and corrective action process shall address the appropriate requirements of the QA program corrective action system and the following elements:

   a. Method(s) for documenting, evaluating, and correcting software problems shall:

      (1) describe the evaluation process for determining whether a reported problem is an error; and
      (2) define the responsibilities for disposition of the problem reports, including notification to the originator of the results of the evaluation.

   b. When the problem is determined to be an error, the method shall provide, as appropriate, for:

      (1) how the error relates to appropriate software engineering elements;
      (2) how the error impacts past and present use of the computer program;
      (3) how the corrective action impacts previous development activities;
      (4) how the users are notified of the identified error, its impact; and how to avoid the error, pending implementation of corrective actions.

2. The methods to be used for reporting operational software problems and taking appropriate action shall be described in implementing procedures which comply with paragraphs above and the requirements of Procedure Manual 1Q, Procedure 15-1, "Control of Nonconforming Items" and Procedure Manual 1B, *Management Requirements and Procedures*, Procedure 4.23, "Corrective Action Program".

I. Software Security Controls

The methods to be used to secure SRS computer systems are described in Procedure Manual 10Q, Computer Security Manual, and the corresponding implementing procedures. The CTF along with the responsible technical authorities shall ensure that the security controls comply with applicable site automated data processing system security requirements. The CTF shall ensure that software generated by WSRC organizations receive reviews for classified or sensitive information in accordance with the requirements of Procedure Manual 7Q, Security Manual.

J. Safety Software Inventory List (SSIL)

The Safety Software Inventory List contains software that meet the requirements of DOE Order 414.1C as listed in Attachment D. Safety software is software used for a Safety System, Safety and Hazard Analysis, Design, Safety Management, or Administrative Controls as detailed in Attachment D. The SSIL will be reviewed, updated and maintained on an annual basis by the QA organization. See Attachment C for SSIL example.

Records

Records shall be controlled in accordance with the requirements of Procedure Manual 1Q, Procedure 17-1, "Quality Assurance Records Management." The following shall be retained as Quality Assurance records based on software classification:

- software classification documentation
- software quality assurance plan(s) or procedure(s)
- software design requirements documentation
- software design documentation
- implementation documentation
- user documentation
- test procedures/plans, test cases, and test results
- evidence of required reviews, review comments and their disposition, and approvals.

References

- For a current list of Source Document references, go to the Standards/Requirements Identification Document (S/RID) webpage accessible through ShRINE
- 10 Code of Federal Regulations (CFR) 71, "Packaging and Transportation of Radioactive Material"
- 10 CFR 830, "Nuclear Safety Management"
- 10 CFR 835, "Occupational Radiation Protection"
- American National Standards Institute (ANSI)/American Society of Quality (ASQ) E4, "Quality Systems for Environmental Data and Technology Programs – Requirements with Guidance for Use"
- American Society of Mechanical Engineers (ASME) NQA-1-2000, Quality Assurance Requirements For Nuclear Facility Applications
- Department of Energy /National Nuclear Security Administration (DOE/NNSA) QC-1, "Weapon Quality Policy QC-1"
- DOE/RW-0333P, "Quality Assurance Requirements and Description (QARD) for the Civilian Radioactive Waste Management Program"
- DOE G 414.1-4, *Safety Software Guide*
- DOE M 470.4-6, *Nuclear Material Control and Accountability (Change 1)*
- DOE O 414.1C, *Quality Assurance*
- DOE P 450.4, *Safety Management System Policy*
- NNSAM56XB, NNSA Development and Production Manual
- Procedure Manual 1B, *Management Requirements and Procedures*, Procedure 4.23, "Corrective Action Program"
- Procedure Manual 7B, *Procurement Management Manual*
- Procedure Manual 12B, *Information Management*, Procedure 1.05, "Software Management"
- Procedure Manual 3E, *Procurement Specification Procedure Manual*
- Procedure Manual 1Q, *Quality Assurance Manual*,
  - Procedure 2-3, "Control of Research and Development Activities"
  - Procedure 2-7, "QA Program Requirements for Analytical Measurement Systems"
  - Procedure 7-3, "Commercial Grade Item Dedication"
  - Procedure 12-1, "Control of Measuring and Test Equipment"
  - Procedure 15-1, Control of Nonconforming Items"
  - Procedure 17-1, "Quality Assurance Records Management"
  - Appendix A, "Glossary of Terms"
- Procedure Manual 7Q, *Security Manual*
- Procedure Manual 10Q, *Computer Security Manual*
- Procedure Manual E7, *Conduct of Engineering and Technical Support*,
  - Procedure 1.30, "Component Numbering System"
  - Procedure 2.05, "Modification Traveler"
  - Procedure 2.25, "Functional Classifications"
  - Procedure 2.31, "Engineering Calculations"
  - Procedure 3.60, "Technical Reports"

Forms

OSR 19-337	Software Classification Document

Attachments

Attachment A.	Graded Approach to Software Classification
Attachment B.	Life Cycle Documentation Requirements Matrix/SQA Graded Approach
Attachment C.	Safety Software Inventory List Template
Attachment D.	Safety Software Inventory List (SSIL) Criteria

Attachment A. Graded Approach to Software Classification (page 1 of 2)

The Life Cycle Documentation associated with these software classifications is provided in Attachment B, "Life Cycle Documentation Requirements Matrix/SQA Graded Approach."

SSC Functional Classification (SC, SS, PS, GS)

- Software that is part of a SSC is classified using

E7 procedures 2.25 & 2.05, then go to Finish Classification Process

Non SSC Classification (A,B,C,D,E)

Classification = "A"

- Software applications that have a DIRECT effect on nuclear safety protection systems that keep exposure to the general public below the off-site regulatory or evaluation guidelines.
    - Include software running on hardware that has no direct output connections to an SSC, but whose output is used without further review or evaluation as a DIRECT input to the functioning of an SSC. (Note: Few instances of Level "A" software are expected at SRS.)

If "A" Go to Finish Classification Process

Classification = "B"

- Software applications whose failure to properly function may have an indirect effect on nuclear safety protection systems or toxic materials hazard systems that are used to keep nuclear or toxic material hazard exposure to the general public and workers below regulatory or evaluation guidelines
- Software applications whose results are used to make decisions that could result in death or serious injury or are part of the evaluation in accident analyses.

However the software application may meet a criteria for lower classification if:

- the software output is used for defense-in-depth as determined by the safety analysis.
    - Classification can be changed to "C"
- The software is used to perform tasks related to hazards routinely encountered in general industry and construction and for which national consensus codes, standards or site programs exist to guide safe design and operation.
    - Classification can be changed to "D"

If Classification complete, Go to Finish Classification Process

Classification = "C"

- Software applications whose failure to perform as expected would not affect nuclear safety but would have an unacceptable impact by causing loss of:
    - greater than $2 Million Dollars production investment value and/or recovery cost
    - primary program capabilities in excess of six months.
- Software applications important to continued operations of the business and that which is used to support decisions regarding operating activities.
- Software applications used to comply with regulatory laws, environmental permits or regulations and/or commitments to compliance.
- Software applications required by the SRS Emergency Plan for environmental monitoring or for communications with Local, State and Federal Government agencies.
    - Include software that is used in nuclear and non-nuclear facilities for trending and analysis of operational data; or to provide to operation or maintenance information to management in support of decisions regarding operating activities, causing an unacceptable impact.
    - Include software whose output is used for defense-in-depth as determined by the safety analysis.

If "C" Go to Finish Classification Process

Attachment A. Graded Approach to Software Classification (page 2 of 2)

Classification = "D"

- Software applications important to the day-to-day administration of the business but whose failure to perform as intended will not adversely affect the safety or reliability of operations or will not result in losses exceeding $2 Million Dollars or result in a six month loss of program capabilities.
    - Include software used to perform tasks related to hazards that are routinely encountered in general industry and construction and for which national consensus codes, standards or site programs exist to guide safe design and operation.

If "D" Go to Finish Classification Process

Classification = "E"

- Software that is within scope of this procedure but does not meet the criteria specified in the above classification levels.

Finish Classification Process

- Document Classification OSR-19-337 or E7 3.60
- Perform CQF Review including sending copy of OSR 19-337 to the Performing Entity QA Manager for software to be included in the Safety Software Inventory List
- Perform Independent Review of Classification as required
- Implement the SQA requirements associated with the software classifications provided in Attachment B, "Life Cycle Documentation Requirements Matrix "

Attachment B. Life Cycle Documentation Requirements Matrix/SQA Graded Approach (page 1 of 1)

Graded Approach to Software Quality Assurance (QAP 20-1 Flow)

Software Classification Level SSC			**SC / SS**			**PS / GS**			**NA**
Software Classification Level Non-SSC			**A / B**			**C / D**			**E**
NQA-1 2000	**ASME NQA-1 2000 Subpart 2.7 / QAP 20-1 Requirements R = Required G = Graded O = Optional**	**QAP 20-1 Section**	**Developed**	**Existing**	**Purchased**	**Developed**	**Existing**	**Purchased**	**E Software**
100	Software Classification	B	R	R	R	R	R	R	R
101	SQA Procedures / Plans	C	R	R	R	R	R	R	O
101	**Life Cycle Phases**	D							
200	Requirements	D.2	R	R	R	R/G	G	G	O
401/ 402	Design	D.3	R	G	G	G	G	G	O
403	Implementation	D.4	R	G	G	G	G	G	O
403	Testing	D.5	R	G	R	G	G	G	O
404	Installation & Acceptance	D.6	R	R	R	R/G	G	R/G	O
405/ 406	Operations & Maintenance	D.7	R	R	R	G	G	G	O
407	Retirement	D.8	R	R	R	G	G	G	O
	SQA Actions								
203	Configuration Control	E	R	R	R	R/G	R/G	R/G	O
302	Evaluation	F	NA	R	G	NA	R	G	O
301	Dedication of Commercial Grade	G	NA	NA	R	NA	NA	G	O
204	Problem Reporting & Corrective Action	H	R	R	R	G	G	G	O
405	Software Security Controls	I	R	R	R	R	R	R	O
	Safety Software Inventory List (SSIL)	J	CTF / Design Agency / CQF / software owner review software for safety software determination (Attachment D)						

Notes: NA = Not Applicable

R/G = Required for PS and C Classifications. Graded for GS and D Classifications

Attachment C. Safety Software Inventory List Template (page 1 of 1)

Current version is posted on the QA webpage via Shrine

The Safety Software Inventory List (SSIL) is a listing of software that meets the requirements for safety software as defined by DOE Order 414.1C as identified in Attachment D. The SSIL consist of the following categories of software:

- Safety System Software
- Safety and Hazard Analysis Software and Design Software
- Safety Management and Administrative Controls Software.

The software on the SSIL is identified by the CTF/Design Authority/CQF. The CQF organizations provide the safety software information for their organizations to the Performing Entity QA Manager who is responsible for the maintenance of the SSIL.

Safety Software Inventory List (Typical)
Safety System Software List

Software Name / ID Number	SSIL Type	Author	Classification	Design Agency / Authority	STE/Sponsor/Owner	SQAP #
Safety System PLC	Safety System Software	Rosemont	SS	P&CS / Engineering	Operations	B-SQP-G-*****
Tank Top Loading	Safety and Hazard Analysis Software -	Vince Gombotz	B	-	Tank Farm Engineering	B-SQP-G-*****
GTStrudl	Design Software	GA Tech	B	-	Joe Bagadoughnuts	B-SQP-G-*****
SRPP	Safety Management Software	Univ. of South Carolina	B	-	WSMS	B-SQP-G-*****
TEF Worker Protection	Administrative Control Software	-	SS	P&CS	DP Operations	B-SQP-G-*****

Attachment D. Safety Software Inventory List (SSIL) Criteria (page 1 of 1)

As part of the software classification process, the Software Owner/Design Authority determines if the software is safety software that should be included on the SSIL. The Design Agency/CTF recommends and ensures documentation is complete. The safety software determination is based on the requirements identified in DOE Order 414.1C and the accompanying guide DOE G 414.1-4. These requirements are identified in the definitions below. Software that meets these requirements will be added to the SSIL. When the CQF approves the Software Classification Document (OSR 19-337) for software identified as safety software, they are to provide a copy of the form to the Performing Entity QA Manager so that software can be added to the SSIL.

Software that should be considered Safety Software and included in the SSIL

(1) **Safety System Software** - Software for a nuclear facility that performs a safety function as part of a structure, system, or component and is cited in either (a) a DOE approved documented safety analysis or (b) an approved hazard analysis per DOE P 450.4, Safety Management System Policy, and the DEAR clause.

(2) **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 a structure, system, or component (SSC) but helps to ensure the proper accident or hazards analysis of nuclear facilities or an SSC that performs a safety function.

(3) **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.