NUCLEAR ENERGY INSTITUTE
NUCLEAR INFORMATION TECHNOLOGY STRATEGIC LEADERSHIP

GUIDANCE DOCUMENT TO IMPLEMENT POLICY FOR SOFTWARE QUALITY ASSURANCE IN THE NUCLEAR POWER INDUSTRY

NITSL-SQA-2005-02

Revision 1

January 15, 2009
ACKNOWLEDGEMENTS

Nuclear Information Technology Strategic Leadership (NITSL) is a NEI Community of Practice (COP) that provides a forum for information technology decision-makers and professionals to sponsor, empower, and promote overall coherence of activities, which support the Nuclear Information Technology community. NITSL sponsored the working group to develop the policy and implementing guidance document to coordinate a consistent direction for Software Quality Assurance.

NITSL STEERING COMMITTEE, 2005
Stephen A Deskevich, Duke Energy
Brooks M. Boylston, Progress Energy
Robert Haverkamp, Southern California Edison
Mark Draxton, Constellation Energy
Randall Tate, Exelon Corp
Greg Przyjemski, NITSL Program Manager

NITSL Steering Committee Sponsor, 2005
Mark Draxton, Constellation Energy

NITSL EXECUTIVE COMMITTEE, 2009
Steve Baleno, South Carolina Electric & Gas
Keith Cooke, Energy Northwest
Glen Frix, Duke Energy
Bill Higgins, Southern California Edison
Gregory Przyjemski, Total Resource Mgt
Ruth Todd, INPO
Bradley Yeates, Southern Company

NITSL Steering Committee Sponsor, 2009
Bill Higgins, Southern California Edison

Team Members
Cynthia Broadwell, Team Lead, Progress Energy
Rich Buell, Entergy
Daniel Bierbrauer, Constellation Energy
James Jones, Duke Energy
Alan Lord, AmerenUE

Independent Reviewers
Steve DeGange, Duke Energy
Tom Duke, Duke Energy
Bob Quay, Energy Northwest
Bill Higgins, Southern California Edison
John Prehn, AEP
Keith Morrell, Westinghouse Savannah River Site
Dave Valley, STP
Rick Hackett, Arizona Public Service Company

Revision 1 Reviewers
Cynthia Broadwell, Progress Energy
Garry Brown, Entergy Nuclear
Pete Grondin, First Energy Corporation
John Hansen, Exelon, Inc.
Nolan Henrich, Jr., Tennessee Valley Authority
Bill Higgins, Southern California Edison
Eric Jurotich, Southern Company
Keith Morrell, Savannah River Solutions
Mary McKenna, Electric Power Research Institute
Donald Nowicki, Florida Power and Light
P. Lynne Valdez, Arizona Public Service Company
1.0 Purpose

This guidance document provides suggestions for implementing NITSL-SQA-2005-01, POLICY FOR SOFTWARE QUALITY ASSURANCE IN THE NUCLEAR POWER GENERATION INDUSTRY and it replaces the former Nuclear Utility Software Management Group (NUSMG) Guidance Documents. Guidance contained herein recommends acceptable processes and controls to develop, implement, and manage Software Quality Assurance Programs. Individual utilities may implement these recommendations as written, or may customize them to be compatible with their existing programs in meeting the intent of NITSL-SQA-2005-02.

2.0 Scope

2.1 This document applies to software used in safety systems covered by 10CFR50 Appendix B.

2.2 Additional management policies may exist to control other processes.

2.3 Certain exclusions to the SQA program and procedures may be identified by management if the exclusions are identified and under the control of another quality process.

3.0 Definitions


3.1 Dedication

The process of confirming that a commercial grade item is acceptable for performing a nuclear safety-related function. Dedication occurs after receipt and after completion of verification and/or validation activities, but before the item is made available for use as a basic component.

3.2 Development

Process by which computer programs (including source code) are written or modified.

3.3 Software

Software includes computer programs, procedures, operating system, applications, rules, and documentation.

3.4 Software Quality Assurance

The program that establishes controls for the development, procurement, operation, use, maintenance, and retirement of software commensurate with its importance to nuclear safety.
3.5 **Graded Approach**

The selective assignment of the quality assurance elements that the software must comply with based on its assigned quality classification. This is determined by the evaluation of the functional process(es) the software supports.

3.6 **Software Life Cycle**

The period of time that begins when a software product is conceived and ends when the software is no longer available for use. Although typical life cycle phases are identified below, many types of software life cycles may be defined. The specific program should identify the software life cycle and accompanying life cycle phases to be used. Phases associated with software management may include the following:

- **Planning** – evaluation of options and coordination of activities to assure successful deployment of software
- **Requirements** – specific and measurable characteristics that describe the intended use and performance of software
- **Design** – the collection of information (requirements, architecture, etc.) that define software Implementation
- **Implementation** – the process of translating the Design into software components
- **Integration** – the process of combining software components with other software and/or systems and to reduce the introduction of undesirable characteristics (errors, anomalies, hazards, security threats)
- **Validation** – specific and measurable tests to demonstrate the software meets its Requirements
- **Installation** – placing software into the operational computing environment, documenting its baseline configuration, and performing final acceptance testing
- **Operation and Maintenance** – on-going use of software and control of changes
- **Retirement** – activities associated with the permanent removal of software from its operational computing environment

3.7 **Software Quality Assurance (SQA)**

The program that establishes quality controls for the development, procurement, operation, use, maintenance, and retirement of software commensurate with its importance to nuclear safety.
4.0 Software Quality Assurance Responsibilities

4.1 Senior Management should establish the policy for software quality assurance.

4.2 SQA Program Owner

4.2.1 Develop, administer, and monitor the program

4.2.2 Define the extent of program interface with plant QA program elements (for example: procurement, records and document control)

4.2.3 Develop procedures and guidance documents that implement programmatic requirements

4.2.4 Provide consultant services and guidance on the program to developers, end users and management

4.2.5 Develop program training

4.2.6 Address error management

4.2.7 Develop and maintain Software QA process including:

1. Software quality level classifications

2. Software life cycle controls

4.2.8 Develop self-assessment activities to assure the program is consistently implemented and identify issues for improvement
4.3 Software Owner

4.3.1 Comply with program requirements to address software life cycle phases

4.3.2 Evaluate, determine, and document software classification

4.3.3 Assure that the actual use of the software is consistent with the classification

4.3.4 Ensure the software is included in plant configuration management when applicable

4.3.5 Assure the software meets its functional requirements

4.3.6 Assure QA elements and activities applied to the development and maintenance of software are commensurate with the intended use

4.3.7 Evaluate, approve, and control software change requests

4.3.8 Support procurement activities by:

1. Including software classification and requirements in applicable procurement documents

2. Assure that the software is procured from or modified by a qualified contractor or vendor, as applicable.

3. Require software error notification in procurement documents, when appropriate

4. Assure license and maintenance agreements are maintained and controlled

5. Control contracts for software development

6. Establish and participate in commercial grade dedication of software, as warranted

4.3.9 Assure that software documentation is developed and maintained

4.3.10 Approve software documentation

4.3.11 Develop software training, as appropriate

4.3.12 Assure completion of acceptance testing

4.3.13 Disposition software error notices and corrective actions

4.3.14 Maintain software records as directed by records management processes
4.3.15 Ensure a disaster recovery plan is defined, as appropriate, for software critical to plant safety and operation.

4.3.16 Ensure cyber security protections are defined, as appropriate, for software critical to plant safety and operation.

4.3.17 Determine whether encryption and authentication measures are required, if appropriate

4.3.18 Other responsibilities to consider for Digital Upgrades:
   1. Document owner of the software
   2. Identify support responsibilities required for the project
   3. Perform Owner review(s) of vendor activities and products
   4. Specify Mock-ups, test lab, simulator needs
   5. Provide input to Training for qualification for operation and maintenance

4.4 End User

4.4.1 Obtain training to use software

4.4.2 Maintain qualification to use software, if required by plant training program

4.4.3 Use software in accordance with its classification

4.4.4 Report software errors and problems

4.4.5 Comply with computing environment security practices and procedures

4.4.6 Participate in assigned user acceptance testing

4.5 Information Technology Support

4.5.1 Maintain and control the operational computing environment

4.5.2 Coordinate changes to the operational computing environment

4.5.3 Implement security measures for the operational computing environment

4.5.4 Communicate schedule for operational computing environment changes to end users when software is affected

4.5.5 Establish control of license and maintenance agreements

4.5.6 Provide software development support, as requested
4.5.7 Support and/or implement disaster prevention and recovery

4.5.8 Provide routine backup services on mainframe and network servers

4.5.9 Provide software problem resolution or resolution support

4.6 Purchasing and Procurement

4.6.1 Assure that SQA procurement requirements are included in purchasing documents

4.6.2 Include vendor and contractor error notification requirements in purchasing and/or procurement documents, as appropriate

4.6.3 Assure appropriate handling, storage and shipment of media and computing hardware and software products

4.6.4 Assure vendor is qualified to provide the requested products and services.
   1. Vendors may be qualified to provide 10CFR50 Appendix B services and products. Generally a vendor with this qualification is included on plant vendor supplier lists.
   2. Vendors may also be qualified to provide “Commercial” products
      a. If used in 10CFR50 Appendix B processes, dedication is required.
      b. Many commercial-off-the-shelf products are being used when deploying digital upgrades for non-safety related application.

5.0 SQA Program Elements

Software quality assurance procedures should define requirements and controls to be applied to software consistent with its importance to safety (the graded approach to quality). Attachment 1 provides a recommended approach for the application of SQA Program Element controls to the various levels of software classification. Each program element is linked to the 10CFR50 Appendix B criteria supported. These should include:

5.1 Organization Responsibility (criterion I)

Add responsibilities reflecting those outlined in section 4.0 to implementing procedures.

5.1.1 Define the software quality assurance program.

1. Scope and applicability of the software quality assurance program.

2. Authority and responsibility for implementing and governing software quality assurance.

3. Oversight of the Software Quality Assurance Program to ensure regulatory compliance.
5.1.2 Define accountability for ownership of the software quality assurance procedure.

1. Establish ownership of the software quality assurance program in applicable policies and procedures.

2. Establish roles and responsibilities for users, information technology personnel, software owners, supervisors, managers, etc.

5.1.3 Define accountability for ownership of the software.

1. Establish ownership of the software in applicable procedures.

2. Establish roles and responsibilities for users, information technology personnel, software owners, supervisors, managers, etc.

5.2 SQA Program (criterion II)

5.2.1 Software Quality Classification

Criteria to classify software important to nuclear safety should be established and reflected in quality levels using a graded approach. Optional sub-categories may be included in SQA programs similar to those levels suggested below.

1. High Impact

Software that has a direct active effect on the ability of a Safety-Related structure, system or component (SSC) to perform its intended safety functions.

Software used for the design of SSC that assures the SSC meets its intended design basis safety function as defined in the nuclear license documents without using alternate methods to verify the results.

2. Medium Impact

Software used to assess the ability of SSC to meet its intended safety function.

Software used to monitor operation and control functions of plant SSC.

3. Low Impact

Software used to support activities that have no direct impact on nuclear operations, design or license commitments but may be used to monitor compliance or optimize performance.

4. Other

Software not included in the above classifications.
5.2.2 Define SQA program interfaces within the overall Quality Assurance Program.

1. Software Quality Assurance and control of software is an integral part of activities that affect quality. SQA activities interface with procurement, engineering, configuration control, document control, records management, corrective action programs, and others.

2. SQA interfaces to consider when developing software may include existing design control and configuration control process. The development of software could also be a stand-alone process governed exclusively by the SQA life cycle documentation process. See Section 5.3.

3. SQA interfaces to consider when procuring software are described in section 5.4 and 5.6.

4. SQA interfaces to document management (See Section 5.7) are important to ensure proper control of software life cycle documentation. Consideration should be given to require revision control and distribution for the following documents for software installed in plant systems and safety related software:
   a. Software Requirements Specification (SRS)
   b. Requirements Traceability Matrix (RTM)
   c. Software Design Description (SDD)
   d. Software Design Specification (SDS)
   e. Software Verification and Validation Plan (SVVP)

5. The SQA interfaces with the QA Records program. Software life cycle documentation should be a QA Record for software that is installed in plant digital systems, software that is safety related or important to nuclear safety, and software that supports regulatory requirements or technical specifications. See Section 5.7.

6. SQA interfaces to the plant configuration control process should be carried out for software installed in plant digital systems. See Section 5.8.

7. SQA interfaces with the Corrective Action program for error management. See Section 5.9.

8. SQA interfaces with plant and corporate audit organizations. The SQA program should be included on plans for periodic review. See Section 5.10.
5.2.3 Define SQA program indoctrination and training requirements for software governed by the SQA policy.

1. Personnel who procure, develop and use software that is installed in plant digital systems should receive training in accordance with the licensee’s training program.

2. Formal documented qualifications of individuals who receive SQA training should be considered.

3. Training of these individuals should be monitored and provided at such intervals deemed appropriate to support qualification.

4. Training should be provided for those who procure, develop and use software that is safety related or important to nuclear safety to assure qualification, as appropriate.

5. SQA Training for software programmers should be provided.

6. Lower tier “awareness” type training should be considered for others.

5.3 Design, Development, Modification and Testing (Software Life Cycle) (criteria III, XI)

5.3.1 Apply controls to software, according to its quality classification, from the time specifications are approved until the system is retired.

1. Develop documentation in accordance with applicable Design processes.

2. Software used to support business processes, installed on computers/servers that are not considered a plant structure, system or component, should be designed, developed, modified and tested in accordance with software quality assurance procedures.

   a. These procedures should provide instruction for the development of software life cycle documentation for software that is custom designed. This includes in-house development, or development by vendors or consultants.

   b. The procedures should include instruction for establishing configuration control at installation and when software is modified.

3. Software installed in digital plant computing systems that are classified as a plant structure, system or component should be designed, developed, modified and tested in accordance with engineering design procedures and software quality assurance procedures.

4. Testing of plant digital computing systems should include simulator testing and the creation of a mock-up identical to target implementation to ensure intended safety function is accomplished.
5. When developing software, either in-house or software that is custom developed by a supplier, software life cycle documentation and testing must be carried out.

a. Life cycle documents generated during the development phase include one or more of the following based on the software quality level:

- Functional Requirements Document (FRD)
- Software Requirements Specification (SRS)
- Requirements Traceability Matrix (RTM)
- Software Design Description (SDD)
- Software Design Specification (SDS)
- Software Verification and Validation Plan (SVVP)
- Software Verification and Validation Report (SVVR)

b. NRC Regulatory Guides 1.168 – 1.173 endorse several IEEE standards that provide guidance on a content and format of software life cycle documents that is acceptable to the staff.

c. Develop software life cycle documentation to match the software quality level and degree of change. The graph below represents an example from one facility and is provided for information only.
d. One of the most effective methods to document software is by using a matrix that identifies software requirements and links the individual requirement to one or more tests. These tests should be specific and measurable to a sufficient degree to demonstrate that the requirement was met. Refer to Attachment 3 for an example RTM. The RTM may be modified to implement only the sections required for a given software quality classification.
5.4 Procurement (criterion IV, VII, XIII)

5.4.1 Develop procurement documents that specify the software Requirements to assure the vendor meets the design intent.

1. Software that is purchased commercial off the shelf (COTS) or “shrink wrap” should have adequate test to assure it meets the expectations of the requesting organization. Many times this requires the organization to develop a functional requirements document and acceptance tests to demonstrate their expectations are met.

2. During the proposal stage, especially for customized software, it is important to ensure the vendor or consultant understands and commits to the expected software life cycle deliverables. Organizations should develop a functional requirements document to assist with vendor and customer understanding of expectations.

3. Vendor life cycle deliverables should be similar to those generated by in-house staff (refer to Section 5.3) although some may be considered proprietary and may not be included in the deliverable. At a minimum, the vendor should supply test cases that demonstrate the software meets the expected requirements.

4. The vendor or consultant should understand and have experience producing life cycle documents so that a proposal adequately covers the effort required to generate these documents.

5. If planning to dedicate software, the purchasing organization should:
   a. Identify and document, in the functional requirements, those critical design characteristics the software must possess to accomplish the intended safety functions
   b. For plant digital computing systems this may include simulator testing and the creation of a mock-up identical to target implementation to ensure intended safety function is accomplished
   c. Mock-up testing may be carried out at the vendor facility and/or the utility
   d. Establish critical acceptance characteristics and, in test documentation, demonstrate the safety functions the software must perform are acceptably implemented

6. Refer to Attachment 2 which lists items to consider when purchasing software.
5.4.2 Establish conditions in procurement documents to assure the control of quality by vendor or consultant when providing software and/or services.

1. Vendors or consultants who provide software that is included in safety related plant systems or that support other 10 CFR 50 Appendix B activities are required to maintain an SQA program equivalent to that maintained by the utility.

2. These programs are audited by one or more utilities for adequacy. Most often information regarding the adequacy of the vendor or consultant QA program, to implicitly include software, is maintained on a list of approved vendors to provide 10CFR50 products.

3. Vendors or consultants may also be commercially qualified for non-safety related software products.

5.4.3 Specify any special shipping, storage, and handling requirements for media or firmware in procurement documents.

1. Consider temperature, humidity, electromagnetic interference, etc.

5.5 Procedures and Instructions (criterion V)

5.5.1 Procedures governing software should be reviewed, approved, and controlled.

1. SQA procedures should:
   a. Identify how requests for quality software are processed and establish required approvals for various software activities
   b. Identify the software quality level categories that implement the graded approach to quality for software
   c. Define life cycle requirements based on the software quality level category
   d. Contain templates, based on software quality level, for development phase life cycle documentation format and content
   e. Provide guidance for managing software errors
   f. Contain QA Record storage requirements for software life cycle documentation

2. Procedures and instructions governing software should be included in the nuclear document control program.

3. SQA Procedures should be included in plant operation or administrative manuals.
4. Activities directed by SQA may be contained in one or more procedures.

5. In order to support SQA interfaces to other programs, references to software activities should be included in interfacing procedures including but not limited to procurement, engineering, corrective action, and document management, and non-SQA information technology procedures, such as disaster recovery procedures.

6. Evaluate existing procedures and corporate policies to leverage requirements and prevent duplication.

7. SQA procedures should be limited to governing activities that address software control requirements.

8. Procedures or processes that govern associated topics including but not limited to computing hardware, electronic data management, disaster recovery, computer security, etc. should be referenced by SQA procedures.

9. SQA procedures should identify roles and responsibilities for those who develop and/or procure, own, maintain, and use quality software.

10. Roles and responsibilities for SQA Program ownership, maintenance and oversight should also be established (refer to 5.1).

5.6 Reviews and Inspections (criteria III, X, XIV)

5.6.1 Define measures to assure that the software meets procurement specification requirements before the software is accepted.

1. Acceptance tests should be carried out at the vendor’s facility and the customer’s facility to demonstrate software performs as expected.

2. For complex plant computing software, periodic vendor surveillance should be considered.

5.6.2 Establish controls to assure reviews, testing and inspection of software and documentation are performed prior to use.

1. Receipt inspection criteria should be established to assure software acceptability prior to final acceptance.
2. When software is dedicated, additional steps may be required to carry out receipt inspection. For example, conditional release may be necessary for testing to demonstrate acceptance of critical characteristics on target computer environments prior to final acceptance.

5.7 Documentation and Records (criteria VI and XVII)

5.7.1 Define document control and records management requirements for software life cycle documentation.

1. Software life cycle documentation that should be considered life time records at a minimum, includes but is not limited to the following:
   a. If software is internally developed, development phase life cycle documents (refer to 5.7.2) should be stored as a QA record.
   b. If software is externally provided, life cycle documents should be stored as a QA record to the extent possible.
   c. Procurement documents such as purchase orders and contracts
   d. Installation and Acceptance Test
   e. Qualification documents
   f. Independent reviews and other approvals should be included, as required

5.7.2 Develop documentation sufficient to prove the quality of the software during its life cycle.

1. Development phase life cycle documents should be generated based on the software quality level

2. Internally developed software should include:
   a. At a minimum the FRD, SRS, RTM, SDD, SDS, SVVP, and SVVR should be generated for software supporting 10CFR50 Appendix B activities
   b. The RTM is highly recommended and should be a requirement for internally developed software.

3. Externally provided software should include:
   a. At a minimum the SRS, SVVP and SVVR should be provided for High and Medium Impact software.
b. The RTM is highly recommended and should be implemented to the extent possible for vendor or contractually supplied software

5.8 Configuration Management (criterion VIII)

5.8.1 Describe software to assure unique identification necessary to maintain configuration. (name, version, platform).

1. Establish baselines for software to define the basis for further development, allow control of configuration items, and permit traceability between configuration items.

   a. Formal control of life cycle activities including development of upgrades for existing software, installation of new versions of software, or other activities that affect software version increments essential to establishing baselines for software that defines the basis for further development, allows control of configuration items, and permits traceability between configuration items.

2. Use configuration activities to control and document changes to baselines.

   a. Formal control can be achieved through existing design control processes for software installed in plant digital systems.

   b. Formal control of software installed in business computing systems generally rely on paper forms like checklists, accepted commercial life cycle control methodology, or database processes established in enterprise management systems or homegrown databases.

   c. Essential attributes of these controlling processes include review and approval activities, life cycle and/or procurement documentation, documented installation and testing activities, and storage of documentation in QA Records.

3. To assure unique identification necessary to maintain configuration. (name, version, platform) plant configuration control process should be used for software installed in plant digital systems.

4. Configuration control of this classification of software should be managed just as it is with other plant equipment.
5. While there is no regulatory requirement to maintain configuration control of software that is operational in the business computing environment, some SQA programs may maintain configuration of name, version, platform, ownership, etc. in master databases or catalogs. Most often this classification of software supports regulatory requirements, is business important, or supports plant availability.

6. Configuration of this category of software may also be tracked through controlled “scripts” used for pushing software to workstations. This method is most effective when workstations are designed and the image is controlled.

5.9 Provisions for Error Management (criteria XV, XVI)

5.9.1 Upon notification or discovery of a software defect, determine whether the defect meets the threshold of the corrective action program.

1. The evaluation to determine whether the software problem is a user or operational problem or a software defect may include the initiation of a Helpdesk case.

2. Validated software errors are required to be included in the corrective action program.

3. Provisions should be included in the SQA program to notify the software users of errors, and document any error avoidance and/or remedial action as applicable.

5.9.2 If the identified condition is a result of user or operational error, include in the corrective action program for trending and OE purposes.

5.9.3 If the identified condition is an enhancement, it should be handled through the business planning process and may include the initiation of a software change request, or modification request.

5.10 Audits (criterion XVIII)

5.10.1 The audit checklist, Section III – Software, suggested by the Nuclear Procurement Issues Committee (NUPIC) forms an acceptable method to audit software suppliers. This checklist addresses the following items in greater detail.

1. Measures are established to control software throughout the life cycle.

2. Life cycle activities are adequately and effectively reviewed.

3. Acceptance testing is adequate.

4. Changes are controlled commensurate with the original software development.
5. Measures are established and implemented for the procurement of software either safety-related or commercial grade.

6. Measures are established and implemented to assure that software errors and failures from both internal and external sources are identified, documented, resolved, evaluated, assessed for impact on past and present applications, and resolved.

7. Measures are established and implemented to assure that software is adequately packaged, marked, stored and shipped.

5.10.2 Internal oversight organizations including plant assessment organizations and corporate oversight organizations should also carry out audit of SQA programs.

5.10.3 The SQA program owner should perform periodic self-assessment of the adequacy of the SQA program as well as the level of compliance by the users.

6.0 References

6.1 10CFR50, Appendix B - Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants

6.2 NUREG 0800, Section 7, BTP HICB-14, Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems

6.3 IEEE 610.12-1990, Software Engineering Terminology


6.5 Regulatory Guide 1.168, Verification, Validation, Reviews, and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants


7.0 Superseded NUSMG Documents


7.4  Quality Software Procurement Guideline (1994)
### ATTACHMENT 1

**Suggested Minimum Requirements for Software Based Upon Classification**

NOTE: Individual utilities may implement these recommendations as written, or may customize to be compatible with their existing programs in meeting the intent of Policy Document NITSL-SQA-2005-01

<table>
<thead>
<tr>
<th>SQA Program Elements</th>
<th>HIGH</th>
<th>MED</th>
<th>LOW</th>
<th>OTHER</th>
<th>10CFR Appendix B Criteria*</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Criteria I - Organization</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintain Organization to Maintain SQA Program</td>
<td>R</td>
<td></td>
<td></td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>Require Owner for Software</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>I</td>
</tr>
<tr>
<td>Audit and Inspect the Program</td>
<td>R</td>
<td></td>
<td></td>
<td>I</td>
<td></td>
</tr>
<tr>
<td><strong>Criteria II – Quality Assurance Program</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graded Approach - control activities (software) affecting SSCs consistent with importance to nuclear safety</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>II</td>
</tr>
<tr>
<td>Classify Software</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>II</td>
</tr>
<tr>
<td>Training</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td></td>
<td>II</td>
</tr>
<tr>
<td>Control Environment</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>II</td>
</tr>
<tr>
<td>• For Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>XI</td>
</tr>
<tr>
<td>• Controlling Quality</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Controlled Infrastructure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-Assessment Program</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td></td>
<td>II</td>
</tr>
<tr>
<td><strong>Criteria III – Design Control</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design standards are specified and deviations controlled</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td></td>
<td>III</td>
</tr>
<tr>
<td>Verify and check adequacy of design</td>
<td>R</td>
<td>S</td>
<td></td>
<td></td>
<td>III</td>
</tr>
<tr>
<td>• Verification Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Independent Design Reviews and Testing</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td>III</td>
</tr>
<tr>
<td>• Independent Design Verification</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Design Reviews and testing</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td>III</td>
</tr>
<tr>
<td>Design changes subject to same controls as original design</td>
<td>R</td>
<td>S</td>
<td></td>
<td></td>
<td>III</td>
</tr>
<tr>
<td>Qualification Testing</td>
<td>R</td>
<td>S</td>
<td></td>
<td></td>
<td>III</td>
</tr>
<tr>
<td>Acceptance Functional Test</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>III</td>
</tr>
<tr>
<td>Benchmark Testing</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td>III</td>
</tr>
<tr>
<td>Computer Environment Qualification</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td>III</td>
</tr>
<tr>
<td>Surveillance Testing</td>
<td>R</td>
<td>S</td>
<td></td>
<td></td>
<td>III</td>
</tr>
<tr>
<td>Regression Test</td>
<td>R</td>
<td>S</td>
<td></td>
<td></td>
<td>III</td>
</tr>
<tr>
<td><strong>Criteria IV – Procurement Document Control</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control Procurement Document</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td></td>
<td>IV</td>
</tr>
<tr>
<td>Vendor Provide Evidence of Quality</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td></td>
<td>IV</td>
</tr>
<tr>
<td>Specify Quality Requirements</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td></td>
<td>IV</td>
</tr>
<tr>
<td>• Error Notifications</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Criteria V – Instructions, Procedures and Drawings</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manuals/Instructions/Procedures</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>SQA Program Elements</td>
<td>HIGH</td>
<td>MED</td>
<td>LOW</td>
<td>OTHER</td>
<td>10CFR Appendix B Criteria*</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>-------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Description of Functions</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td>V</td>
</tr>
<tr>
<td>Requirement Specification</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Design Specification</td>
<td>R</td>
<td>S</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>V&amp;V Test Plan</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Acceptance Criteria</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>V&amp;V Test Results</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td>V</td>
</tr>
<tr>
<td>Installation Instructions/Plan</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td>V</td>
</tr>
<tr>
<td><strong>Criteria VI - Document Control</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Control Process</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td>VI</td>
</tr>
<tr>
<td>Source Code—(Can escrow if purchased)</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td></td>
<td>VI</td>
</tr>
<tr>
<td>Life Cycle Document Control</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td></td>
<td>VI</td>
</tr>
<tr>
<td>Version Control</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td></td>
<td>VI, VIII, XIV</td>
</tr>
<tr>
<td><strong>Criteria VII – Control of Purchased Material, Equipment and Services</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receipt Inspection</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td></td>
<td>VII</td>
</tr>
<tr>
<td>• Software Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>• Validation Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>XV</td>
</tr>
<tr>
<td>Vendor Qualification</td>
<td>R</td>
<td>R</td>
<td></td>
<td></td>
<td>VII</td>
</tr>
<tr>
<td>Vendor Audit</td>
<td>R</td>
<td>R</td>
<td></td>
<td></td>
<td>VII</td>
</tr>
<tr>
<td><strong>Criteria VIII – Identification of Materials, Parts, and Components</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inventory</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>VIII</td>
</tr>
<tr>
<td>Executable Programs</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td></td>
<td>VIII</td>
</tr>
<tr>
<td><strong>Criteria IX – Control of Special Processes</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Embed Software on Devices</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td></td>
<td>IX</td>
</tr>
<tr>
<td><strong>Criteria X – Inspection</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Define Inspection/Test points</td>
<td>R</td>
<td>R</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Installation Checkout Test</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td>X</td>
</tr>
<tr>
<td>Hardware/Software Integration testing</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Validation Testing</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td>X</td>
</tr>
<tr>
<td><strong>Criteria XI – Test Control</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualified Personnel</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td></td>
<td>XI</td>
</tr>
<tr>
<td>Documented Test</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td>XI</td>
</tr>
<tr>
<td>Acceptance Criteria</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td>XI</td>
</tr>
<tr>
<td><strong>Criteria XII – Control of Measuring and Test Equipment</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calibration Testing</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td></td>
<td>XII</td>
</tr>
</tbody>
</table>

Legend:  R – Required  S – Suggested
<table>
<thead>
<tr>
<th>SQA Program Elements</th>
<th>HIGH</th>
<th>MED</th>
<th>LOW</th>
<th>OTHER</th>
<th>10CFR Appendix B Criteria*</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Criteria XIII – Handling, Storage and Shipping</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Magnetic Medium and Digital Information Control</td>
<td>R</td>
<td>S</td>
<td>S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Define storage controls</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Govern by Procedure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Criteria XIV – Inspection, Test, and Operating Status</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Computer System Security | R    | R   | S   | S     | III
|                         |      |     |     |       | XIV
|                         |      |     |     |       | XV
| Classify Product | R    | S   | S   | S     | XIV
| Identify Production/Release Status | R    | S   | S   | S     | XIV
| Identify Approved Use | R    | S   | S   | S     | XIV
| **Criteria XV – Nonconforming Materials, Parts, or Components** |      |     |     |       |                           |
| Identify non-conformance during | R    | R   | S   | S     | XV
| • Installation Checkout Test |      |     |     |       |                           |
| • Hardware/Software Integration testing |      |     |     |       |                           |
| • Validation Test |      |     |     |       |                           |
| Tag non-conforming items | R    | R   | S   | S     | XV
| **Criteria XVI – Corrective Action** |      |     |     |       |                           |
| Corrective Action Program | R    | R   | S   | S     | XVI
| Software or Digital System Error |      |     |     |       |                           |
| • Identify |      |     |     |       |                           |
| • Notify |      |     |     |       |                           |
| • Track |      |     |     |       |                           |
| • Disposition |      |     |     |       |                           |
| **Criteria XVII – Quality Assurance Records** |      |     |     |       |                           |
| Retain Records | R    | R   | S   | S     | XVII
| **Criteria XVIII - Audits** |      |     |     |       |                           |
| Audit | R    | R   | S   | S     | XVIII
| • Independent |      |     |     |       |                           |
| • Trained and Qualified Assessors |      |     |     |       |                           |
ATTACHMENT 2
PROCUREMENT REQUEST

Vendor Information
Name: _____________________________________________________
Address: _____________________________________________________
Telephone: _____________________________________________________
Contact: _____________________________________________________

Vendor Qualifications
☐ Vendor on Approved Supplier’s List for Appendix B software
☐ Add Vendor to Approved Supplier’s List for Appendix B software
☐ Purchase CGI and Dedicate Internally
☐ Commercial
☐ Other _____________________________________________________

Software Information
Software name: _______________ Version: __________
Software Quality Level: ________________________________
Shipping, Handling, and Storage requirements ___________________________
_________________________________________________________________

Deliverables
Deliverables Required Delivered Version #
☐ Software Version Upgrade ______________________
☐ Source Code ______________________
☐ Software ______________________
☐ Software Life Cycle Documents ______________________
☐ Test Instructions ______________________
☐ Test Cases ______________________
☐ Electronic Test Files ______________________
☐ 10CFR50 Appendix B Support Agreement ______________________
☐ Maintenance Agreement ______________________
☐ Technical Support Agreement ______________________
☐ Error Notification ______________________
☐ Error Correction Updates ______________________
Other ______________________

Receipt Inspection
Receipt Tag Requirements: _______________________________________
Dedication Method: _______________________________________________
Post Issuance Test Requirements: _________________________________
_________________________________________________________________

NOTE:
1. All electronic files, inputs, and results shall be delivered in a format consistent with standard software products, unless non-standard deliverable is approved
2. Specify in the procurement documentation that the supplier shall notify the purchaser of non-conformances identified in the product for Safety Related purchases.
3. Specify in the procurement documentation that updates to correct software errors shall be provided to the purchaser for vendor maintained software.
## Requirements Traceability Matrix Examples

### Requirements Traceability Matrix – High Impact Software

<table>
<thead>
<tr>
<th>SRS</th>
<th>SDD</th>
<th>TEST PLAN</th>
<th>TEST REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement</td>
<td>Design Entity</td>
<td>Test Activity</td>
<td>Expected Result</td>
</tr>
</tbody>
</table>

Acceptance Criteria:
- **High**: Critical requirement that is mandatory and must be met
- **Medium**: Non-critical element that will be included if possible
- **Low**: Deferred for later development

### Requirements Traceability Matrix – Medium Impact Software

<table>
<thead>
<tr>
<th>SRS</th>
<th>TEST PLAN</th>
<th>TEST REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement</td>
<td>Test Activity</td>
<td>Expected Result</td>
</tr>
</tbody>
</table>

Acceptance Criteria:
- **High**: Critical requirement that is mandatory and must be met
- **Medium**: Non-critical element that will be included if possible
- **Low**: Deferred for later development

### Requirements Traceability Matrix – Low Impact Software

<table>
<thead>
<tr>
<th>FRD</th>
<th>TEST REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement</td>
<td>Pass</td>
</tr>
</tbody>
</table>

Acceptance Criteria:
- **High**: Critical requirement that is mandatory and must be met
- **Medium**: Non-critical element that will be included if possible
- **Low**: Deferred for later development