





Software Quality Assurance

## SQA Checklists for Audits

WP-2020-SAF-QA-SQA-001

Prepared By:   
Patrick V. Auer, Lawrence Livermore National  
Laboratory

June 23, 2020  
Date

Approved By:   
Vicki L. Pope, Lawrence Livermore National  
Laboratory

June 23, 2020  
Date

## **SUMMARY:**

The NQA-1-based checklists and spreadsheet serve as a baseline which the user can customize for specific tasks, scopes of work, or supplier programs. The key to using the checklists and spreadsheet follows:

- The “EFCOG software checklist” is a 10-item, high-level summary of the overall NQA-1 requirements and would require software QA and engineering subject matter experts (SMEs) with significant experience in assessments and software QA to complete the objective evidence portion of the lines of inquiry.
- The SMEs would need to be familiar with the specific software QA requirements being assessed.
- The software QA program documents and procedures are reviewed in advance (i.e., the supplier sends their procedures and program manual to the audit team prior to the on-site visit) and are documented separately using the “EFCOG Safety Software QA Program Review Checklist”.
- The audit team reviews the supplier’s QA program documents to become familiar with the supplier’s software QA program
- The excel spreadsheet is a way to tailor the requirements for different types of software (e.g., developed, utility, acquired) and the resulting requirements would be what the SME addresses during the assessment.
- These checklists are just for the software portion of the NQA-1 program for ease of review
- The checklists are written to NQA-1 2000 through 2008, where “otherwise acquired software” is qualified in accordance with Para 302 of Subpart 2.7
- The checklists would need to be revised for an audit involving NQA-1 2008/2009a where otherwise acquired software is “dedicated” by the supplier or in-house in accordance with Subpart 2.14

### Contributions by:

James Hylko, Paducah

## **PURPOSE:**

The purpose or goal for this white paper is to standardize checklists used to perform audits of suppliers where software is used or developed. The standardized software checklists help ensure supplier audits are documented consistently, and prior to placement on the Master Supplier List (MSL) or used at participating DOE sites.

## **SCOPE:**

The scope of the task was to develop a standard method for documenting NQA-1 supplier audits, including the initial program review, when the supplier’s scope of work includes use, qualification, or development of software.

Background: software audit documentation was not consistent in audits performed and placed on the MSL or its predecessor, the Master Approved Supplier List (MASL). The use of software subject matter experts on audit teams was not consistent, resulting in less than adequate evaluations of supplier QA programs which involved software.

## **DEFINITIONS:**

NONE.

**NARRATIVE:**

In general, there was an observation that evaluations of supplier QA programs were not consistent and, in some cases, less than adequate. In addition, supplier audit teams were not consistently using appropriate SMEs to evaluate specialized areas such as software.

**CONCLUSION:**

The attached audit checklists and spreadsheets can be used by EFCOG members to create a standard method of documenting supplier audits when the supplier's scope of work includes use, qualification, or development of software.

In addition, the checklists could be used at individual sites to be more consistent in how software-related audits or assessments are documented and promote sharing of these reports between sites (i.e., outside MSL).

**REFERENCES:**

NQA-1-2000 to 2008, Requirements 3, 11, and Subpart 2.7

**ATTACHMENTS:**

Attachment 1, EFCOG Software Assessment Checklist (Blank)

Attachment 2, EFCOG Software QA Program Review Checklist (NQA-1 2008 subpart 2.7 – Blank)

Attachment 3, EFCOG NQA-1 2008, subpart 2.7 Software Quality Worksheet (file attached, excel spreadsheet)



EFCOG NQA-1 2008  
Subpart 2.7 Softwar

Attachment 1 – EFCOG Software Assessment Checklist

**Subpart 2.7 - Software Quality**

Item Number	Requirement/ Implementing Document	Line of Inquiry	Objective Evidence	Results
1		Verify management practices are documented in the Supplier's Software Quality Program.		
2		Verify that the software is listed on the Supplier's Software inventory or baseline.		
3		Verify related support software is subject to adequate configuration management controls.		
4		Confirm software design and/or usage requirements adequately defined and documented.		
5		Confirm that the supplier adequately manages and documents software acquisition activities.		
6		Verify the software design, verification and implementation processes are adequately defined and documented.		
7		Confirm that the Supplier controls for software distribution, installation, removal and retirement are implemented.		
8		Verify test plans and results (V&V) are documented and traceable to design and/or usage requirements. How are the software test records retained?		
9		Verify that users of the software are provided instructions, training, and guidance. Are the applicable in-use testing procedures followed?		
10		Does the supplier adequately manage software errors, including impact on past and present use of active and retired software? How does the supplier ensure notices are received, evaluated, and communicated?		

Attachment 1 – EFCOG Software Assessment Checklist

**Subpart 2.7 - Documents Reviewed**

Number	Title	Revision	Date

**Subpart 2.7 - Software Quality Table**

Software Program (Name, No., Rev. / Date)	Program End Use (e.g. Design, Production, Calibration, Acceptance)	Verification / Validation	Method / Procedure To Control Issuance of Changes and / or Error Notices

## Safety Software QA Program Review Checklist (NQA-1 2008)

### How to use this form:

1. List QA Program Manual section or paragraph in "Requirement Established" if that is where the requirement is established.
2. List procedure, program manual section, instruction, flow chart, documented information, or other implementing mechanism. If the requirement is established or only discussed in the "Methodology" document, list it in the "Requirement Established" column.
3. Results: identify if the documents meet the requirement and provide an adequate method to implement the requirement. (Sat, Unsat, or N/A)

### Results:

Section	Results	Comments
3 - Design Control		
11 - Test Control		
19 - Subpart 2.7, Quality Assurance Requirements for Computer Software for Nuclear Facility Applications		

### Reviewed by:

\_\_\_\_\_  
QA Program Reviewer

\_\_\_\_\_  
Signature

\_\_\_\_\_  
Title

\_\_\_\_\_  
Date

## Safety Software QA Program Review Checklist (NQA-1 2008)

### Section 3 - Design Control

Requirement	Requirement Established	Methodology Described	Results	Comments
<p><b>100 Basic</b></p> <p>The design shall be defined, controlled, and verified. Design inputs shall be specified on a timely basis and translated into design documents. Design interfaces shall be identified and controlled. Design adequacy shall be verified by individuals other than those who designed the item or computer program. Design changes shall be governed by control measures commensurate with those applied to the original design.</p>				
<p><b>401 Use of Computer Programs</b></p> <p>To the extent required in paras. 401(a) and (b) of this Requirement, computer program acceptability shall be preverified or the results verified with the design analysis for each application. Preverified computer programs shall be controlled in accordance with the requirements of this Standard.</p> <p>(a) The computer program shall be verified to show that it produces correct solutions for the encoded mathematical model within defined limits for each parameter employed.</p> <p>(b) The encoded mathematical model shall be shown to produce a valid solution to the physical problem associated with the particular application.</p>				
<p><b>402 Documentation of Design Analysis</b></p> <p>Documentation of design analyses shall include the following:</p> <p>(a) the objective of the analyses</p> <p>(b) design inputs and their sources</p> <p>(c) results of literature searches or other applicable background data</p> <p>(d) assumptions and indication of those assumptions that must be verified as the design proceeds</p>				

## Safety Software QA Program Review Checklist (NQA-1 2008)

### Section 3 - Design Control

Requirement	Requirement Established	Methodology Described	Results	Comments
<p>(e) identification of any computer calculation, including identification of the computer type, computer program name, and revision, inputs, outputs, evidence of or reference to computer program verification, and the bases (of reference thereto) supporting application of the computer program to the specific physical problem</p> <p>(f) review and approval</p>				
<p><b>700 Interface Control</b></p> <p>Interface controls shall include assignment of responsibility and establishment of procedures among participating design organizations for review, approval, release, distribution, and revision of documents involving design interfaces. Design information transmitted across interfaces shall identify the status of the design information or document provided, and identify incomplete items that require further evaluation, review, or approval. Where it is necessary to initially transmit design information orally or by other informal means, the transmittal shall be confirmed promptly by a controlled document.</p>				
<p><b>800 Software Design Control</b></p> <p>The requirements of section 800 apply to computer software design control and shall be used instead of section 200, Design Input; section 300, Design Process; section 500, Design Verification; and section 600, Change Control. Part II, Subpart 2.7, Quality Assurance Requirements for Computer Software for Nuclear Facility Applications, provides work practice requirements to implement the requirements of this paragraph.</p>				
<p><b>801 Software Design Process</b></p> <p>The software design process shall be documented, approved by the responsible design organization, and controlled. This process shall include the activities described in paras. 801.1 through 801.5 of this Requirement.</p>				



## Safety Software QA Program Review Checklist (NQA-1 2008)

### Section 3 - Design Control

Requirement	Requirement Established	Methodology Described	Results	Comments
<p><b>801.1 Identification of Software Design Requirements</b></p> <p>Software design requirements shall be identified and documented and their selection reviewed and approved. The software requirements shall identify the operating system, function, interfaces, performance requirements, installation considerations, design inputs, and any design constraints of the computer program.</p>				
<p><b>801.2 Software Design</b></p> <p>The software design shall be documented and shall define the computational sequence necessary to meet the software requirements. The documentation shall include, as applicable, numerical methods, mathematical models, physical models, control flow, control logic, data flow, process flow, data structures, process structures, and the applicable relationships between data structures and process structures. This documentation may be combined with the documentation of the software design requirements, or the computer program listings resulting from implementation of the software design.</p>				
<p><b>801.3 Implementation of the Software Design</b></p> <p>The software design shall be translated into computer program(s) using the programming organization's or design organization's programming standards and conventions.</p>				
<p><b>801.4 Software Design Verification</b></p> <p>Software design verification shall be performed by a competent individual(s) or group(s) other than those who developed and documented the original design, but who may be from the same organization. This verification may be performed by the originator's supervisor, provided</p>				

## Safety Software QA Program Review Checklist (NQA-1 2008)

### Section 3 - Design Control

Requirement	Requirement Established	Methodology Described	Results	Comments
<p>(a) the supervisor did not specify a singular design approach or rule out certain design considerations and did not establish the design inputs used in the design, or</p> <p>(b) the supervisor is the only individual in the organization competent to perform the verification.</p> <p>Cursory supervisory reviews do not satisfy the intent of this Standard.</p> <p>The results of verification shall be documented with the identification of the verifier indicated. Software verification methods shall include any one or a combination of design reviews, alternate calculations, and tests performed during computer program development. The extent of verification and the methods chosen are a function of the complexity of the software, the degree of standardization, the similarity with previously proved software, and the importance to safety.</p>				
<p><b>801.5 Computer Program Testing</b></p> <p>Computer program testing shall be performed and shall be in accordance with Requirement 11.</p>				
<p><b>802 Software Configuration Management</b></p> <p>Software configuration management includes, but is not limited to configuration identification, change control, and status control. Configuration items shall be maintained under configuration management until the software is retired.</p>				
<p><b>802.1 Configuration identification.</b></p> <p>A software baseline shall be established at the completion of each activity of the software design process. Approved changes created subsequent to a</p>				

## Safety Software QA Program Review Checklist (NQA-1 2008)

### Section 3 - Design Control

Requirement	Requirement Established	Methodology Described	Results	Comments
<p>baseline shall be added to the baseline. A baseline shall define the most recently approved software configuration.</p> <p>A labeling system for configuration items shall be implemented that</p> <p>(a) uniquely identifies each configuration item</p> <p>(b) identifies changes to configuration items by revision</p> <p>(c) provides the ability to uniquely identify each configuration of the revised software available for use</p>				
<p><b>802.2 Configuration Change Control.</b></p> <p>Changes to software shall be formally documented. The documentation shall include</p> <p>(a) a description of the change</p> <p>(b) the rationale for the change</p> <p>(c) the identification of affected software baselines</p> <p>The change shall be formally evaluated and approved by the organization responsible for the original design, unless an alternate organization has been given the authority to approve the changes. Only authorized changes shall be made to software baselines. Appropriate verification activities shall be performed for the change. The change shall be appropriately reflected in documentation, and traceability of the change to the software design requirement shall be maintained. Appropriate acceptance testing shall be performed for the change.</p>				
<p><b>802.3 Configuration Status Control.</b></p> <p>The status of configuration items resulting from software design shall be maintained current. Configuration item changes shall be controlled until</p>				

## Safety Software QA Program Review Checklist (NQA-1 2008)

### Section 3 - Design Control

Requirement	Requirement Established	Methodology Described	Results	Comments
they are incorporated into the approved product baseline. The controls shall include a process for maintaining the status of changes that are proposed and approved, but not implemented. The controls shall also provide for notification of this information to affected organizations.				
<b>900 Documentation and Records</b>  Design documentation and records shall include not only final design documents, such as drawings and specifications, and revisions to those documents, but also documentation that identifies the important steps in the design process, including sources of design inputs that support the final design.				

### Section 11 - Test Control

Requirement	Requirement Established	Methodology Described	Results	Comments
<b>100 Basic</b>  Tests required to collect data such as for siting or design input, to verify conformance of an item or computer program to specified requirements, or to demonstrate satisfactory performance for service shall be planned and executed. Characteristics to be tested and test methods to be employed shall be specified. Test results shall be documented and their conformance with test requirements and acceptance criteria shall be evaluated.				
<b>200 Test Requirements</b>				

## Safety Software QA Program Review Checklist (NQA-1 2008)

### Section 11 - Test Control

Requirement	Requirement Established	Methodology Described	Results	Comments
<p><i>(a)</i> Test requirements and acceptance criteria shall be provided or approved by the responsible design organization. Required tests, including, as appropriate, prototype qualification tests, production tests, proof tests prior to installation, construction tests, preoperational tests, operational tests, and computer program tests such as software design verification, factory acceptance tests, site acceptance tests, and in-use tests shall be controlled.</p> <p>Required tests shall be controlled under appropriate environmental conditions using the tools and equipment necessary to conduct the test in a manner to fulfill test requirements and acceptance criteria. The tests performed shall obtain the necessary data with sufficient accuracy for evaluation and acceptance.</p>				
<p><i>(b)</i> Test requirements and acceptance criteria shall be based upon specified requirements contained in applicable design documents, or other pertinent technical documents that provide approved requirements.</p>				
<p><i>(c)</i> If temporary changes to the approved configuration of a facility are required for testing purposes, approval by the design authority is required prior to performing the test.</p>				
<p><b>400 Computer Program Test Procedures</b></p> <p>The requirements of Section 400 of Requirement 11 apply, instead of Section 300, Test Procedures, to testing of computer programs, and as appropriate, the computer hardware and operating system. apply, instead of section 300, Test Procedures, to testing of computer programs, and as appropriate, the computer hardware and operating system.</p> <p><i>(a)</i> Computer program test procedures shall provide for demonstrating the adherence of the computer program to documented requirements. For those computer programs used in design activities, computer program test procedures shall provide for assuring that the computer program produces</p>				

## Safety Software QA Program Review Checklist (NQA-1 2008)

### Section 11 - Test Control

Requirement	Requirement Established	Methodology Described	Results	Comments
correct results. For those computer programs used for operational control, computer program test procedures shall provide for demonstrating required performance over the range of operation of the controlled function or process. The procedures shall also provide for evaluating technical adequacy through comparison of test results from alternative methods such as hand calculations, calculations using comparable proven programs, or empirical data and information from technical literature.				
<i>(b)</i> In-use test procedures shall be developed and documented to permit confirmation of acceptable performance of the computer program in the operating system. In-use test procedures shall be performed after the computer program is installed on a different computer, or when there are significant changes in the operating system. Periodic in-use manual or automatic self-check in-use tests shall be prescribed and performed for those computer programs in which computer program errors, data errors, computer hardware failures, or instrument drift can affect required performance.				
<p><b>500 Test Results</b></p> <p>Test results shall be documented and evaluated by a responsible authority to ensure that test requirements have been satisfied. Test results for design qualification tests and software design verification shall be evaluated by the responsible design organization.</p>				
<p><b>600 Test Records</b></p> <p>Test records shall be established and maintained to indicate the ability of the item or computer program to satisfactorily perform its intended function or to meet its documented requirements. Test records vary depending on the test type, purpose, and application, but shall contain the following information, as a minimum, for the specified application identified in paras. 601 and 602.</p>				

## Safety Software QA Program Review Checklist (NQA-1 2008)

### Section 11 - Test Control

Requirement	Requirement Established	Methodology Described	Results	Comments
<p><b>602 Computer Program Test Records</b></p> <p><i>(a) Verification Test Records</i></p> <p>(1) computer program tested</p> <p>(2) computer hardware tested</p> <p>(3) test equipment and calibrations, where applicable</p> <p>(4) date of test</p> <p>(5) tester or data recorder</p> <p>(6) simulation models used, where applicable</p> <p>(7) test problems</p> <p>(8) results and applicability</p> <p>(9) action taken in connection with any deviations noted</p> <p>(10) person evaluating test results</p> <p><i>(b) In-Use Test Records</i></p> <p>(1) computer program tested</p> <p>(2) computer hardware tested</p> <p>(3) test equipment and calibrations, where applicable</p> <p>(4) date of test</p> <p>(5) tester or data recorder</p> <p>(6) acceptability</p>				

## Safety Software QA Program Review Checklist (NQA-1 2008)

### Subpart 2.7, Quality Assurance Requirements for Computer Software for Nuclear Facility Applications

Requirement	Requirement Established	Methodology Described	Results	Comments
<p><b>100 General</b></p> <p>Subpart 2.7 provides requirements for the acquisition, development, operation, maintenance, and retirement of software. The appropriate requirements of this Subpart shall be implemented through the policies, procedures, plans, specifications, or work practices, etc., that provide the framework for software engineering activities. Subpart 2.7 supplements the requirements of Part 1 and shall be used in conjunction with applicable Requirements of Part 1 when and to the extent specified by the organization invoking the subpart.</p>				
<p><b>101 Software Engineering</b></p> <p>The scope of software engineering activities* include the following elements, as appropriate:</p> <ul style="list-style-type: none"> <li>a) Software acquisition method(s) for controlling the acquisition process for software and software services;</li> <li>b) Software engineering method(s) used to manage the software life-cycle activities;</li> <li>c) Application of standards, conventions, and other work practices that support the software life cycle; and</li> <li>d) Controls for support software used to develop, operate, and maintain computer programs.</li> </ul>				
<p><b>200 General Requirements</b></p> <p>The following general requirements shall be applied to the software engineering elements described in para. 101 of this Subpart.</p>				
<p><b>201 Documentation</b></p> <p>The appropriate software engineering elements, described in para. 101 of this Subpart, shall define the baseline documents that are to be maintained</p>				



## Safety Software QA Program Review Checklist (NQA-1 2008)

### Subpart 2.7, Quality Assurance Requirements for Computer Software for Nuclear Facility Applications

Requirement	Requirement Established	Methodology Described	Results	Comments
as records, in accordance with Part I, Requirement 17. Although multiple documentation requirements are specified within this Subpart, they can be provided as separate or as combined documents.				
<p><b>202 Review</b></p> <p>The appropriate software engineering elements, described in para. 101 of this Subpart, shall define the control points and associated reviews. Reviews of software shall assure compliance with the approved software design requirements. Although multiple review requirements are specified within this Subpart, the reviews may be performed and documented separately or combined, as appropriate, to the defined software engineering method. The following two reviews are required:</p> <p>a) One review shall consider the requirements related to the activities of preparing the computer program for acceptance testing. This review can be combined with or be part of the software design verification.</p> <p>b) The other review shall provide assurance of the satisfactory completion of the software development cycle including acceptance testing. This review can be combined with or be part of software design verification. Individual(s) familiar with the design detail and the intended use of the computer program shall be included in the review.</p>				
Reviews shall identify the participants and their specific review responsibilities. Documentation of review comments and their disposition shall be retained until they are incorporated into the updated software. Comments not incorporated and their disposition shall be retained until the software is approved for use. 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.				
Tests performed in support of a review can be used to complement acceptance testing. The tests and test results shall be included in the acceptance testing documentation. Such tests shall be subjected to the				

## Safety Software QA Program Review Checklist (NQA-1 2008)

### Subpart 2.7, Quality Assurance Requirements for Computer Software for Nuclear Facility Applications

Requirement	Requirement Established	Methodology Described	Results	Comments
same criteria as the acceptance tests. These tests do not substitute for performing the comprehensive, end of development, acceptance test.				
<p><b>203 Software Configuration Management</b></p> <p>In addition to the requirements of Part, I Requirement 3, software configuration management activities shall include the following:</p> <p>a) The appropriate software engineering elements, described in para. 101 of this Subpart, shall identify when configuration baselines are to be established. Configuration items to be controlled shall include, as appropriate:</p> <ol style="list-style-type: none"> <li>1. Documentation (e.g., software design requirements, instructions for computer program use, test plans, and results);</li> <li>2. Computer programs (e.g., source, object, back-up files); and</li> <li>3. Support software.</li> </ol> <p>b) The software configuration change control process shall include:</p> <ol style="list-style-type: none"> <li>1. Initiation, evaluation, and disposition of a change request;</li> <li>2. Control and approval of changes prior to implementation; and</li> <li>3. Requirements for retesting and acceptance of the test results.</li> </ol>				
<p><b>204 Problem Reporting and Corrective Action</b></p> <p>a) Method(s) for documenting, evaluating, and correcting software problems shall:</p> <ol style="list-style-type: none"> <li>1. Describe the evaluation process for determining whether a reported problem is an error or other type of problem (e.g., user mistake); and</li> <li>2. Define the responsibilities for disposition of the problem reports, including notification to the originator of the results of the evaluation.</li> </ol> <p>b) When the problem is determined to be an error, the method shall provide, as appropriate, for:</p> <ol style="list-style-type: none"> <li>1. How the error relates to appropriate software engineering elements;</li> <li>2. How the error impacts past and present use of the computer program;</li> </ol>				

## Safety Software QA Program Review Checklist (NQA-1 2008)

### Subpart 2.7, Quality Assurance Requirements for Computer Software for Nuclear Facility Applications

Requirement	Requirement Established	Methodology Described	Results	Comments
<p>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.</p> <p>The problem reporting and corrective action process shall address the appropriate requirements of Part I, Requirement 16.</p>				
<p><b>300 Software Acquisition</b></p> <p>Software acquisition includes software or software services procured in accordance with Part I, or otherwise acquired for use in activities within the scope of Part I.</p>				
<p><b>301 Procured Software and Software Services</b></p> <p>Part I, Requirements 4 and 7 for items and services shall be applied to the procurement of software and software services. The Purchaser shall be responsible for the appropriate requirements of this Subpart upon acceptance of the software or related item (e.g., programmable device). 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.</p>				
<p><b>302 Otherwise Acquired Software</b></p> <p>Software that has not been previously approved under a program consistent with this Standard for use in its intended application (e.g., freeware, shareware, procured commercial off-the-shelf, or otherwise acquired software), shall be evaluated in accordance with the requirements of this Subpart. The software shall be identified and controlled prior to evaluation. The evaluation, specified by this section, shall be performed and documented to determine adequacy to support operation and maintenance</p>				

## Safety Software QA Program Review Checklist (NQA-1 2008)

### Subpart 2.7, Quality Assurance Requirements for Computer Software for Nuclear Facility Applications

Requirement	Requirement Established	Methodology Described	Results	Comments
<p>and identify the activities to be performed and the documentation that is needed.</p> <p>This determination shall be documented and shall identify as a minimum:</p> <ul style="list-style-type: none"> <li>a) Capabilities and limitations for intended use;</li> <li>b) Test plans and test cases required to demonstrate the capabilities within the limitations; and</li> <li>c) Instructions for use within the limits of the capabilities.</li> </ul> <p>Exceptions from the documentation requirements of this Subpart and the justification for acceptance shall be documented.</p> <p>The results of the above evaluation and the performance of the actions necessary to accept the software, shall be reviewed and approved. The resulting documentation and associated computer program (s) shall establish the current baseline.</p> <p>Revisions to previously baseline software received from organizations not required to follow this Subpart shall be evaluated in accordance with this Element.</p>				
<p><b>400 Software Engineering Method</b></p> <p>Software engineering method(s) shall be documented.</p> <p>The selected software engineering method shall ensure that software life cycle activities are planned and performed in a traceable and orderly manner. The appropriate requirements of Part I, Requirement 3 shall be met.</p>				

## Safety Software QA Program Review Checklist (NQA-1 2008)

### Subpart 2.7, Quality Assurance Requirements for Computer Software for Nuclear Facility Applications

Requirement	Requirement Established	Methodology Described	Results	Comments
<p><b>401 Software Design Requirements</b></p> <p>Software design requirements shall address technical and software engineering (i.e., para. 101 of this Subpart) requirements. Software design requirements shall be traceable throughout the software life cycle</p>				
<p><b>402 Software Design</b></p> <p>An integral part of software design is the design of a computer program that is part of an overall system. Thus, the software design shall consider the computer program's operating environment. Measures to mitigate the consequences of problems shall be an integral part of the design. These potential problems include external and internal abnormal conditions and events that can affect the computer program.</p>				
<p><b>402.1 Software Design Verification.</b></p> <p>Software design verification shall evaluate the technical adequacy of the design approach and assure internal completeness, consistency, clarity, and correctness of the software design and shall verify that software design is traceable to the software design requirements. Software design verification shall include review of test results. The software design verification shall be completed prior to approval of the computer program for use. The requirements for the software design verification activity shall be documented in the software engineering method</p>				

## Safety Software QA Program Review Checklist (NQA-1 2008)

### Subpart 2.7, Quality Assurance Requirements for Computer Software for Nuclear Facility Applications

Requirement	Requirement Established	Methodology Described	Results	Comments
<p><b>403 Implementation</b></p> <p>The implementation process shall result in software products such as computer program listings and instructions for computer program use.</p> <p>A review shall be performed in accordance with para. 202 of this Subpart</p>				
<p><b>404 Acceptance Testing</b></p> <p>The acceptance testing activity shall demonstrate that the computer program adequately and correctly performs all intended functions (i.e., specified software design requirements). Acceptance testing shall demonstrate, as appropriate, that the computer program:</p> <ul style="list-style-type: none"> <li>a) Properly handles abnormal conditions and events as well as credible failures</li> <li>b) Does not perform adverse unintended functions; and</li> <li>c) Does not degrade the system either by itself, or in combination with other functions or configuration items.</li> </ul>				
<p>Acceptance testing shall be performed prior to approval of the computer program for use. Configuration items shall be under configuration change control prior to starting acceptance testing. Acceptance testing shall be planned and performed for all software design requirements. Acceptance testing ranges from a single test of all software design requirements to a series of tests performed during computer program development. Performance of a series of tests provides assurance of correct translation between activities and proper function of individual modules. Testing shall include a comprehensive acceptance test performed in the operating environment prior to use.</p>				
<p>The test plans, test cases, and test results shall be documented, reviewed, and approved prior to use of the computer program in accordance with Part I, Requirement 11. Observations of unexpected or unintended results shall be documented and dispositioned prior to test result approval.</p>				

## Safety Software QA Program Review Checklist (NQA-1 2008)

### Subpart 2.7, Quality Assurance Requirements for Computer Software for Nuclear Facility Applications

Requirement	Requirement Established	Methodology Described	Results	Comments
<p>The acceptance testing of changes to the computer program shall be subjected to selective retesting to detect unintended adverse effects introduced during the change. Such testing shall provide assurance that the changes have not caused unintended adverse effects in the computer program, and to verify that a modified system(s) or system component(s) still meets specified software design requirements.</p>				
<p><b>405 Operation</b></p> <p>After the software is approved for use and installed in the operating environment, the use of the software shall be controlled in accordance with approved procedures and instructions. These include, as appropriate</p> <p>(a) application documentation (e.g., application log)</p> <p>(b) access control specifications</p> <p>(c) problem reporting and corrective action</p> <p>(d) in-use tests</p> <p>(e) the configuration change control process</p>				
<p><b>406 Maintenance</b></p> <p>The appropriate software engineering elements, as described in para. 101 of this Subpart, shall identify how changes to the software are controlled. Typically,</p> <p>changes are in response to any of the following:</p> <p>(a) enhancement requests from the user community</p> <p>(b) revisions to software based on software design</p>				

## Safety Software QA Program Review Checklist (NQA-1 2008)

### Subpart 2.7, Quality Assurance Requirements for Computer Software for Nuclear Facility Applications

Requirement	Requirement Established	Methodology Described	Results	Comments
<p>requirements</p> <p>(c) changes to the operating environment</p> <p>(d) reported software problems that must be corrected</p>				
<p><b>407 Retirement</b></p> <p>During retirement, support for the software product is terminated, and the routine use of the software shall be prevented</p>				
<p><b>500 Standards, Conventions, and Practices</b></p> <p>As appropriate, the software engineering method, software acquisition method, or both shall establish the need for standards, conventions, and other required work practices to facilitate software life cycle activities (e.g., software design and implementation activities). Standards, conventions, and other required work practices shall be documented.</p>				
<p><b>600 Support Software</b></p> <p>Support software includes software tools and system software. As appropriate, the software engineering method, software acquisition method, or both shall establish the need for software tools.</p>				



## Safety Software QA Program Review Checklist (NQA-1 2008)

### Subpart 2.7, Quality Assurance Requirements for Computer Software for Nuclear Facility Applications

Requirement	Requirement Established	Methodology Described	Results	Comments
<p><b>601 Software Tools</b></p> <p>Software tools shall be evaluated, reviewed, tested, and accepted for use, and placed under configuration control as part of the software development cycle of a new or revised software product. Software tools that do not affect the performance of the software need not be placed under configuration control.</p>				
<p>In cases involving modifications of software products using the software tools, the configuration of the support software associated with that modification shall be managed. Changes to the software tool shall be evaluated for impact on the software product to determine the level of reviews and retesting that will be required.</p>				
<p><b>602 System Software</b></p> <p>System software consists of the on-line computer programs used to provide basic or general functionality and facilitate the operation and maintenance of the application computer program. Examples include:</p> <p>lower level software layers, assemblers, interpreters, diagnostics, and utilities. System software shall be evaluated, reviewed, tested, and accepted for use as part of the software development cycle of a new or revised software product. System software shall be placed under configuration change control. Changes to the system software shall be evaluated for impact on the software product to determine the level of reviews and retesting that will be required.</p>				