|  |  |  |
| --- | --- | --- |
| **Part II, Subpart 2.7, 100, GeneralSubpart 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 I and shall be used in conjunction with applicable Requirements of Part I when and to the extent specified by the organization invoking the Subpart.** | **Part II, Subpart 2.7, 100, General** **This Subpart 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. This Subpart supplements the requirements of Part I and shall be used in conjunction with applicable Requirements of Part I when and to the extent specified by the organization invoking the Subpart.1** | **No change** |
| **Part II, Subpart 2.7, 101, Software EngineeringThe scope of software engineering activities include the following elements, as appropriate: (a) software acquisition method(s) for controlling the acquisition process for software and software services (b) software engineering method(s) used to manage the software life-cycle activities (c) application of standards, conventions, and other work practices that support the software life cycle (d) controls for support software used to develop, operate, and maintain computer programs.** | **Part II, Subpart 2.7, 101, Software Engineering The scope of software engineering activities includes the following elements, as appropriate:*(a)* software acquisition method(s) for controlling the acquisition process for software and software services *(b)* software engineering method(s) used to manage the software life-cycle activities*(c)* application of standards, conventions, and other work practices that support the software life cycle *(d)* controls for support software used to develop, operate, and maintain computer programs.** | **No change** |
| **Part II, Subpart 2.7, 200, General RequirementsThe following general requirements shall be applied to the software engineering elements described in paragraph 101 of this Subpart.** | **Part II, Subpart 2.7, 200, General Requirements The following general requirements shall be applied to the software engineering elements described in para. 101 of this Subpart.** | **No Change** |
| **Part II, Subpart 2.7, 201, DocumentationThe appropriate software engineering elements, described in paragraph 101 of this Subpart, shall define the baseline documents that are to be maintained 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.** | **Part II, Subpart 2.7, 201, Documentation and Records The appropriate software engineering elements, described in paragraph 101 of this Subpart, shall define the baseline documents that are to be maintained 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.** | **Title Change only – No content changed** |
| **Part II, Subpart 2.7, 202, ReviewThe appropriate software engineering elements, described in paragraph 101 of this Subpart, shall define the control points and associated reviews. Reviews of software shall ensure 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: (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. (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. 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 same criteria as the acceptance tests. These tests do not substitute for performing the comprehensive, end of development, acceptance test.** | **Part II, Subpart 2.7, 202, Verification The appropriate software engineering elements, described in paragraph 101 of this Subpart, shall define the control points and associated reviews. Reviews of software shall ensure 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: (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. (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. 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 same criteria as the acceptance tests. These tests do not substitute for performing the comprehensive, end of development, acceptance test.** | **Title Change only – No content changed** |
| **Part II, Subpart 2.7, 203, Software Configuration ManagementIn addition to the requirements of Part I, Requirement 3, software configuration management activities shall include the following: (a) The appropriate software engineering elements, described in paragraph 101 of this Subpart, shall identify when configuration baselines are to be established. Configuration items to be controlled shall include, as appropriate: (1) documentation (e.g., software design requirements, instructions for computer program use, test plans, and results) (2) computer program(s) (e.g., source, object, backup files) (3) support software (b) The software configuration change control process shall include (1) initiation, evaluation, and disposition of a change request (2) control and approval of changes prior to implementation (3) requirements for retesting (e.g. Regression testing) and acceptance of the test results** | **Part II, Subpart 2.7, 203, Software Configuration Management Software configuration management includes, but is not limited to, configuration identification, change control, and configuration status control. Configuration items shall be maintained under configuration management until the software is retired. The appropriate software engineering elements, described in para. 101 of this Subpart, shall identify when configuration baselines are to be established.** | **Revised but no change as Yellow highlighted red text was previously in NQA-1-2008/1a-2009 Part I, Requirement 3, Paragraph 802** |
|  | **Part II, Subpart 2.7, 203.1, Configuration Identification A labeling system for configuration items shall be implemented that** ***(a)* uniquely identifies each configuration item** ***(b)* identifies changes to configuration items by revision** ***(c)* provides the ability to uniquely identify each configuration of the revised software available for use** | **New addition to Section 203 but no change as Yellow highlighted red text was previously in NQA-1-2008/1a-2009 Part I, Requirement 3, Paragraph 802.1, 2nd paragraph.** |
|  | **Part II, Subpart 2.7, 203.2, Configuration Change Control*****(a)* The software configuration change control process shall include*****(1)* initiation, evaluation, and disposition of a change request*****(2)* control and approval of changes prior to implementation*****(3)* requirements for retesting (e.g., regression testing) and acceptance of the test results*****(b)* A software baseline shall be established at the completion of each activity of the software design process. Approved changes created subsequent to a baseline shall be added to the baseline. A baseline shall define the most recently approved software configuration. Configuration items to be controlled as part of the baseline shall include, as appropriate*****(1)* documentation (e.g., software design requirements, instructions for computer program use, test plans, and results)*****(2)* computer program(s) (e.g., source, object, backup files)*****(3)* support software*****(c)* Changes to software shall be formally documented. The documentation shall include*****(1)* a description of the change*****(2)* the rationale for the change*****(3)* the identification of affected software baselines****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.** | **New addition to Section 203 but no change as Yellow highlighted red text was previously in NQA-1-2008/1a-2009 Part I, Requirement 3, Paragraph 802.1 and 802.2 and Part II, Subpart 2.7, 203, Software Configuration Management** |
|  | **Part II, Subpart 2.7, 203.3, Configuration Status Control****The status of configuration items resulting from software design shall be maintained current. Configuration item changes shall be controlled until 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.** | **New addition to Section 203 but no change as Yellow highlighted red text was previously in NQA-1-2008/1a-2009 Part I, Requirement 3, Paragraph 802.3**  |
| **Part II, Subpart 2.7, 204, Problem Reporting and Corrective Action(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 or other type of problem (e.g., user mistake) (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. The problem reporting and corrective action process shall address the appropriate requirements of Part I, Requirement 16.** | **Part II, Subpart 2.7, 204, Problem Reporting and Corrective Action(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 or other type of problem (e.g., user mistake) (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. The problem reporting and corrective action process shall address the appropriate requirements of Part I, Requirement 16.** | **No Change** |
| **Part II, Subpart 2.7, 300, Software AcquisitionSoftware 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.** | **Part II, Subpart 2.7, 300, Software AcquisitionSoftware 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.** | **No Change** |
| **Part II, Subpart 2.7, 301, Procured Software and Software ServicesPart 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** | **Part II, Subpart 2.7, 301, Procured Software and Software ServicesPart 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** | **No Change** |
| **Part II, Subpart 2.7, 302, Otherwise Acquired SoftwarePart I, Requirement 7, and Part II, Subpart 2.14,Quality Assurance Requirements for Commercial Grade Items and Services, shall be applied to the acquisition 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). The acquired software shall be identified and controlled during the dedication process. The dedication process shall be documented and include the following: (a) identification of the capabilities and limitations for intended use as critical characteristics (b) utilization of test plans and test cases as the method of acceptance to demonstrate the capabilities within the limitations (c) instructions for use (e.g., user manual) within the limits of the dedicated capabilities The dedication process shall be documented 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. Subsequent revisions of accepted software received from organizations not required to follow this Subpart shall be dedicated in accordance with this section.** | **Part II, Subpart 2.7, 302, Otherwise Acquired Software****Part I, Requirement 7, and Part II, Subpart 2.14, Quality Assurance Requirements for Commercial Grade Items and Services, shall be applied to acquired software that has not been previously approved under a program consistent with Part I of this Standard for use in its intended application. This includes computer programs not obtained using the procurement requirements of Part I, such as freeware, shareware, and computer programs from corporate repositories. Otherwise acquired computer programs whose results are verified with the design analysis for each application as specified in Part I, Requirement 3, para. 401 are excluded from the requirements of Part II, Subpart 2.14.****Otherwise acquired computer programs shall be identified and controlled during the dedication process. The dedication process shall be documented and include the following:*****(a)* identification of the capabilities and limitations for intended use as critical characteristics*****(b)* utilization of test plans and test cases as the method of acceptance to demonstrate the capabilities within the limitations*****(c)* instructions for use (e.g., user manual) within the limits of the dedicated capabilities****The dedication process documentation and associated computer program(s) shall establish the current baseline.****Subsequent revisions of the software shall be dedicated in accordance with this section.** | **Revised****Yellow highlighted area is partially addressed by the 2008/1a-2009 Part II, Subpart 2.7, Paragraph 302 except for examples used in the 2017 edition.****Yellow highlighted red text is basically addressed in NQA-1-2008/1a-2009 Part I, Requirement 3, Paragraph 401.** |
| **Part II, Subpart 2.7, 400, Software Engineering MethodSoftware engineering method(s) shall be documented. 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.** | **Part II, Subpart 2.7, 400, Software Engineering Method Software engineering method(s) shall be documented. The selected software engineering method shall ensure that software life-cycle activities are planned and performed in a traceable and orderly manner. The software design process shall be documented, approved by the responsible design organization, and controlled. This process shall include the activities described in paras. 401 through 404. Part II, Subpart 2.7, 400, Software Engineering Method** | **Revised but no change as Yellow highlighted red text was previously in NQA-1-2008/1a-2009 Part I, Requirement 3, Paragraph 801.** |
| **Part II, Subpart 2.7, 401, Software Design RequirementsSoftware design requirements shall specify technical and software engineering (i.e., paragraph 101 of this Subpart) requirements, including security features (e.g., vulnerability protection, and cyber-security).2 Identify applicable reference drawings, specifications, codes, standards, regulations, procedures, or instructions that establish software design requirement test, inspection, and acceptance criteria. Security requirements shall be specified commensurate with the risk from unauthorized access or use. Software design requirements shall be traceable throughout the software life cycle.** | **Part II, Subpart 2.7, 401, Software Design Requirements****Software design requirements shall specify technical and software engineering (i.e., para. 101 of this Subpart) requirements, including security features (e.g., vulnerability protection and cybersecurity).3 Identify applicable reference drawings, specifications, codes, standards, regulations, procedures, or instructions that establish software design requirement test, inspection, and acceptance criteria. Security requirements shall be specified commensurate with the risk from unauthorized access or use. The software requirements shall identify the operating system, function, interfaces, performance requirements, installation considerations, design inputs, and any design constraints of the computer program. Software design requirements shall be traceable throughout the software life cycle. Software design requirements shall be identified and documented and their selection reviewed and approved.** | **Revised but no change as Yellow highlighted red text was previously in NQA-1-2008/1a-2009 Part I, Requirement 3, Paragraph 801.1****Footnote reference changed.** |
| **Part II, Subpart 2.7, 402, Software DesignAn 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, as identified through analysis, 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.** | **Part II, Subpart 2.7, 402, Software Design****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, as identified through** **analysis, 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.****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.** | **Revised but no change as Yellow highlighted red text was previously in NQA-1-2008/1a-2009 Part I, Requirement 3, Paragraph 801.2** |
| **Part II, Subpart 2.7, 402.1, Software Design VerificationSoftware design verification shall evaluate the technical adequacy of the design approach and ensure 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.** | **Part II, Subpart 2.7, 402.1, Software Design Verification****Software design verification shall evaluate the technical adequacy of the design approach and ensure 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.****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*****(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*****(b)* the supervisor is the only individual in the organization competent to perform the verification****Cursory supervisory reviews do not satisfy the intent of this Standard.****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 methods chosen are a function of the complexity of the software, degree of standardization, similarity with previously proved software, and importance to safety.** | **Revised but no change as Yellow highlighted red text was previously in NQA-1-2008/1a-2009 Part I, Requirement 3, Paragraph 801.4** |
| **Part II, Subpart 2.7, 403, ImplementationThe implementation process shall result in software products such as computer program listings and instructions for computer program use. A review shall be performed in accordance with paragraph 202 of this Subpart.** | **Part II, Subpart 2.7, 403, Implementation****The software design shall be translated into computer program(s) using the programming organization’s or design organization’s programming standards and conventions.****The implementation process shall result in software products such as computer program listings and instructions for computer program use. A review shall be performed in accordance with para. 202 of this Subpart.** | **Revised but no change as Yellow highlighted red text was previously in NQA-1-2008/1a-2009 Part I, Requirement 3, Paragraph 801.3** |
| **Part II, Subpart 2.7, 404, Acceptance Testing****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 (a) properly handles abnormal conditions and events as well as credible failures (b) does not perform adverse unintended functions (c) does not degrade the system either by itself, or in combination with other functions or configuration item. 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. 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. 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** | **Part II, Subpart 2.7, 404, Acceptance Testing****The acceptance testing activity shall demonstrate that the computer program adequately and correctly performs all intended functions (i.e., specified software design requirements). Computer program tests including, as appropriate, software design verification, factory acceptance tests, site acceptance tests, and in-use tests shall be controlled.****Test requirements and acceptance criteria for computer programs shall be provided by the organization responsible for the use of the computer program and shall include the following, as applicable:*****(a)* Software design verification testing shall demonstrate the capability of the computer program(s) to provide valid results for test problems encompassing the range of documented permitted usage.*****(b)* Computer program acceptance testing shall consist of the process of exercising or evaluating a system or system component by manual or automated means to ensure that it satisfies the specified requirements and to identify differences between expected and actual results in the operating environment.*****(c)* In-use computer programs testing shall demonstrate required performance over the range of operation of the controlled function or process.** | **Revised but no change as Yellow highlighted red text was previously in NQA-1-2008/1a-2009 Part I, Requirement 11, Paragraph 200. (There is some slight wording differences but the requirements are addressed)** |
|  | **Part II, Subpart 2.7, 404.1, Test Coverage** **Acceptance testing shall demonstrate, as appropriate, that the computer program*****(a)* properly handles abnormal conditions and events as well as credible failures*****(b)* does not perform adverse unintended functions*****(c)* does not degrade the system either by itself or in combination with other functions or configuration items****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.** | **New addition but no change as Yellow highlighted red text was previously in NQA-1-2008/1a-2009 Part II, Subpart 2.7, 404.** |
|  | **Part II, Subpart 2.7, 404.2, Test Plans and Procedures****The requirements of this section apply to testing of computer programs and, as appropriate, the computer hardware and operating system.****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 ensuring that the computer program produces 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.****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 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.****The test plans, test cases, and test results shall be documented, reviewed, and approved prior to use of the computer program.****Test procedures or plans shall specify the following, as applicable:*****(a)* required tests and test sequence*****(b)* required ranges of input parameters*****(c)* identification of the stages at which testing is required*****(d)* criteria for establishing test cases*****(e)* requirements for testing logic branches*****(f)* requirements for hardware integration*****(g)* anticipated output values*****(h)* acceptance criteria*****(i)* reports, records, standard formatting, and conventions****Observations of unexpected or unintended results shall be documented and dispositioned prior to test result approval. Test results shall be evaluated by the responsible authority to ensure that test requirements have been satisfied.** | **New addition but no change as Yellow highlighted red text was previously in NQA-1-2008/1a-2009 Part I, Requirement 11, Paragraphs 400and Part II, Subpart 2.7, 404.** |
|  | **Part II, Subpart 2.7, 404.3, Computer Program Test Records****Test records shall be established and maintained to indicate the ability of the computer program to satisfactorily perform its intended function or to meet its documented requirements.****Test records shall include*****(a)* computer program tested, including system software used*****(b)* computer hardware used*****(c)* test equipment and calibrations, where applicable*****(d)* date of test*****(e)* tester or data recorder*****(f)* simulation models used, where applicable*****(g)* test problems*****(h)* results and applicability*****(i)* action taken in connection with any deviations noted*****(j)* person evaluating test results*****(k)* acceptability** | **Revised but no change as Yellow highlighted red text was previously in NQA-1-2008/1a-2009 Part I, Requirement 11, Paragraphs 600 and 602.** |
|  | **Part II, Subpart 2.7, 404.4, Acceptance Testing of Changes****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 shall verify that a modified system(s) or system component(s) still meets specified software design requirements.** | **New addition but no change as Yellow highlighted red text was previously in NQA-1-2008/1a-2009 Part II, Subpart 2.7, 404.** |
| **Part II, Subpart 2.7, 405, OperationAfter 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 (a) application documentation (e.g., application log) (b) access control specifications (c) computer system vulnerability protections (d) problem reporting and corrective action (e) in-use tests (f) the configuration change control process** | **Part II, Subpart 2.7, 405, OperationAfter 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 (a) application documentation (e.g., application log) (b) access control specifications (c) computer system vulnerability protections (d) problem reporting and corrective action (e) in-use tests (f) the configuration change control process** | **No Change** |
| **Part II, Subpart 2.7, 406, MaintenanceThe appropriate software engineering elements, as described in paragraph 101 of this Subpart, shall identify how changes to the software are controlled. Typically, changes are in response to any of the following: (a) enhancement requests from the user community (b) revisions to software based on software design requirements (c) changes to the operating environment and changes to computer system vulnerability protections (d) reported software problems that must be corrected** | **Part II, Subpart 2.7, 406, MaintenanceThe appropriate software engineering elements, as described in paragraph 101 of this Subpart, shall identify how changes to the software are controlled. Typically, changes are in response to any of the following: (a) enhancement requests from the user community (b) revisions to software based on software design requirements (c) changes to the operating environment and changes to computer system vulnerability protections (d) reported software problems that must be corrected** | **No Change** |
| **Part II, Subpart 2.7, 407, RetirementDuring retirement, support for the software product is terminated, and the routine use of the software shall be prevented.** | **Part II, Subpart 2.7, 407, RetirementDuring retirement, support for the software product is terminated, and the routine use of the software shall be prevented.** | **No Change** |
| **Part II, Subpart 2.7, 500, Standards, Conventions, and Other Work PracticesAs 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.** | **Part II, Subpart 2.7, 500, Standards, Conventions, and Other Work PracticesAs 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.** | **No Change** |
| **Part II, Subpart 2.7, 600, Support SoftwareSupport 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** | **Part II, Subpart 2.7, 600, Support SoftwareSupport 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** | **No Change** |
| **Part II, Subpart 2.7, 601, Software ToolsSoftware 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. 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** | **Part II, Subpart 2.7, 601, Software ToolsSoftware 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. 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** | **No Change** |
| **Part II, Subpart 2.7, 602, System SoftwareSystem 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 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** | **Part II, Subpart 2.7, 602, System SoftwareSystem 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 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** | **No Change** |