NRC INSPECTION MANUAL

IQVB

INSPECTION PROCEDURE 35710

QUALITY ASSURANCE INSPECTION OF SAFETY-RELATED SOFTWARE USED FOR DIGITAL INSTRUMENTATION AND CONTROL IN NUCLEAR APPLICATIONS

Effective Date: 04/18/2024

PROGRAM APPLICABILITY: IMCs 2502, 2507, 2508

35710-01 INSPECTION OBJECTIVES

To verify that software used for safety-related digital instrumentation and control (DI&C) systems is developed in accordance with a Quality Assurance Program (QAP) that complies with the requirements of Appendix B, “Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants,” to Title 10 of the *Code of Federal Regulations* (10 CFR) Part 50, “Domestic Licensing of Production and Utilization Facilities.” Specifically, but not limited to:

Criterion II, “Quality Assurance Program,” as it relates to unique aspects of software project management and organizational processes; software quality assurance (QA) processes; software verification and validation processes; and software configuration management processes.

Criterion III, “Design Control,” as it relates to activities that should be specified in design documents, and the design control measures for verifying or checking the adequacy of the design that are unique to software;

Criterion V, “Instructions, Procedures, and Drawings,” as it relates to guidance for I&C activities affecting quality that may warrant specific documented procedures and guidance on the control of I&C documents that prescribe activities affecting quality specific to software;

Criterion XI, “Test Control,” as it relates to aspects of the testing program specific to software; and

Criterion XVI, “Corrective Action,” as it relates to identifying and correcting failures, malfunctions, and anomalies throughout the software lifecycle.

01.01 To verify the system development activities adheres to process requirements for each lifecycle phase; which include translating a conceptual design into system requirements, developing a system design, implementing the design into hardware and software functions, integrating the software and hardware into a DI&C system, qualifying the hardware, testing the software to verify software requirements have been met, and performing integration and system validation testing of the DI&C system to ensure system requirements have been implemented correctly.

01.02 To verify adequate verification and validation activities have been performed and documented for each lifecycle phase.

01.03 This procedure is to be used in combination with the following inspection procedures when applicable: Inspection Procedure (IP) 35017 “Quality Assurance Program Implementation During Construction and Pre-Construction Activities,” and IP 43004, “Inspection of Commercial-Grade Dedication Programs” and IP 65001.22, “Inspection of Digital Instrumentation and Control (DI&C) System/Software Design Acceptance Criteria (DAC) – Related to ITAAC [Inspections, Tests, Analyses and Acceptance Criteria].”

35710-02 INSPECTION REQUIREMENTS

02.01 Verify that DI&C system requirements have been adequately allocated to software, hardware, and architecture requirements. This includes verifying that the vendor has identified which of the quality requirements for the DI&C system are applicable to the software and hardware.

02.02 Verify that the functional and process characteristics of the software have been defined.

02.03 Verify that a formal software lifecycle process has been established and the design outputs of each phase of this lifecycle have been adequately defined.

02.04 Verify that the software lifecycle and design outputs of each lifecycle phase adequately address the following elements of the lifecycle:

- Requirements
- Design
- Implementation
- Integration
- Testing

02.05 Review the implementing instructions, procedures, plans, and policies in place for the QA, verification and validation, and configuration management controls for the development of software to be used in DI&C safety systems.

35710-03 INSPECTION GUIDANCE

03.01 Requirements: The requirements phase consists of developing a description of what the DI&C system must accomplish and the functional characteristics (e.g., accuracy, functionality, reliability, robustness, safety, security, and timing) the DI&C must exhibit. This description is typically documented in system requirements specification. The system requirements are decomposed into software, hardware, and architecture requirements. The process characteristics for software requirements include completeness, consistency, correctness, style, traceability, unambiguity, and verifiability, modifiable and ranked for importance. Assess the requirements phase activities by performing the following:

a. Verify that the requirements documentation specifies the I&C platform, functionality, performance characteristics, interfaces, installation considerations, design constraints, and security constraints. The I&C platform may be microprocessor-based (e.g., programmable logic controllers) or programmable logic-based (e.g., field programmable gate arrays (FPGA)). DI&C systems that use programmable logic-based technology have unique development activities than one that uses a microprocessor-based I&C

platform. For example, FPGAs use high-level description (HDL) language to describe the structure and behavior of electronic circuits and digital logic circuits.

b. Verify that a formal process is documented and implemented to ensure changes to software requirements are evaluated, reviewed, approved, and documented.

c. Select a sample of system requirements to verify the functional characteristics and software development process characteristics are exhibited by the software requirements description documents by performing the following:

   1. Verify that design bases, including applicable regulatory requirements, standards, and codes are adequately translated into system requirements.

   2. Verify that system requirements have been adequately decomposed into software, hardware, and architecture requirements. The software requirements are typically documented in the software requirements specification. The hardware requirements are typically documented in the hardware design description. The architecture requirements are typically documented in the system architecture description.

   3. Verify that there is two-way traceability of the system requirements to the software, hardware, and architecture requirements.

   4. Verify that each software requirement has a unique identifier.

   5. For process characteristics of the software requirements, verify the following:

      (a) The consistency of the software requirements with the DI&C system requirements and the safety-related system design. No requirement should be contradictory to the other requirements.

      (b) The correctness of the software requirements, including the accurate description of functional requirements and operational environment.

      (c) The style of the software requirement descriptions is understandable.

      (d) The software requirement description is unambiguous (i.e., only one interpretation can be derived from the software requirements).

      (e) The software requirement is verifiable.

   6. Verify that changes to the software requirements maintain traceability throughout development activities.

   7. Verify that verification and validation (V&V) reports are adequate to identify any software requirements or development process issues.

   For hardware description language (HDL) programmed devices (e.g., FPGA), the behavioral characteristics of the device are defined in the requirements specification.

03.02 Design: The design phase consists of: (1) translating the design architecture into hardware and software elements, and then mapping of the software into hardware; (2) development of the software and hardware architecture; and (3) decomposing the software and architectural requirements into software design elements. Software design elements may include equations, algorithms, and control logic.

For HDL-programmed devices, architectural specifications and behavioral characteristics are refined into behavioral descriptions. The behavioral description is done by schematic diagram, block diagram, or HDL (e.g., VHDL, Verilog) that used register-transfer-level abstraction to create high-level representation of a circuit.

Evaluate the design activities of the software lifecycle by performing the following:

a. Verify that procedures are implemented to ensure software design specification documentation is produced, reviewed, approved, baselined, updated as necessary, and placed under configuration control.

b. Verify that for each functional characteristic (e.g., accuracy, reliability, robustness, safety, security, timing), the software architecture and design can satisfy the required characteristic. This may include a concurrent review of the hardware architecture, schematics, and drawings.

   For HDL-programmed devices, verify that the architectural design process accounts for top-level design partitioning. The design partitioning should consider design reliability, functionality, traceability, and verifiability. Verify that the functional characteristics are satisfied by the HDL-programmed device attributes. Attributes that should be considered include asynchronous design, metastability, internal resets, phase locked loop locking time, timing constraints, state machines, multiple clock domains, latches, and high fan-out lines.

c. Verify the software development process characteristics for the software architecture and software design exhibit completeness, consistency, style, traceability, and verifiability.

d. Verify that a process is implemented to establish a software baseline at the completion of each design activity.

e. Verify that procedures are implemented to ensure that changes made to the software architecture and design are evaluated, reviewed, approved, and documented. Verify the documentation includes provisions for documenting a description of the change, rationale for the change, identification of the software baseline affected by the change, and status of the change throughout the implementation process.

f. Verify that V&V reports are adequate to identify any software design or development process issues.

   For HDL-programmed devices, verify that the V&V process accounts for HDL device specific V&V activities are performed, such as behavioral simulation and logic-level simulation.

03.03 Implementation: Assess the implementation activities of the software lifecycle by performing the following:

For microprocessor-based software, the implementation phase consists of translating the completed software design into code.

For HDL-programmed devices, a netlist is synthesized from the HDL code that contains the information on how the components should be connected to the programmable logic chip. The netlist is placed and routed to produce a bitstream that can be downloaded on to the programmable logic chip.

a. Verify that implementation activities, such as the development of operation documentation and management of software releases are completed in accordance with a documented implementation plan.

For microprocessor-based software, verify that the implementation activities, such as the creation of an executable code and software unit testing are completed in accordance with a documented implementation plan.

For HDL-programmed devices, verify that the implementation activities, such as placing and routing of the netlist onto the HDL-programmed device. The hardware implementation activities include developing the final board/module design and fabrication. Verify that the final board/module design conforms with schematic diagrams of the board.

b. Verify that procedures are established and implemented for compliance with rules, methods, and standards.

For microprocessor-based software, verify that procedures are established and implemented for compliance with coding rules, methods, and standards.

c. Verify that V&V reports are adequate to identify any software code (or HDL-programmed device equivalent) or development process issues.

For HDL-programmed devices, verify that the functional hardware verifications have been performed to confirm correct board/module functionality in the operational environment.

03.04 Integration: The integration phase consists of combining components into a single system. This may encompass human-machine interfaces (e.g., displays and controls) and communications with other systems (e.g., non-safety-related distributed control systems, maintenance and test systems). The system may have individual channels or divisions with interfaces to interconnect the channels/divisions.

For microprocessor-based software, the integration phase consists of combining software components and hardware components into a single system.

For HDL-programmed devices, the integration phase consists of combining individual devices into a single system.

a. Verify integration activities have been satisfactorily completed.

   1. Verify that the plans and methods for integrating function channels or divisions of software units or HDL-programmed devices are adequately documented. The software integration plan should also identify what is being integrated, define the integration environment, discuss the management of interfaces, and define the integration sequence.

   2. Verify that there are provisions in procedures to ensure the complete integration of all software units and comprised software modules (or HDL-programmed device equivalent) or any other division of functional parts.

   3. Verify the physical, electrical, and communications interfaces of the integrated channel or division have been configured correctly.

b. Verify that integration test activities and tasks, primary test methods and standards, test cases, test coverage, and acceptance criteria are documented in accordance with Section 03.05.

c. Verify that the integration testing accurately reflects the system architecture.

03.05 Testing: The software testing phase consists of unit testing, integration testing, and system validation testing. This includes software unit testing (or HDL-programmed device equivalent), software integration testing (or HDL-programmed device equivalent), hardware integration verification, environmental qualification testing, and system validation testing. Evaluate the testing activities of the system development lifecycle by performing the following:

a. Verify that there are provisions documented in procedures to ensure that all software and hardware requirements are covered by integration and system validation testing.

b. Verify that documentation supporting all testing activities include the following:

   1. Special conditions and controls, equipment, tools, and instrumentation needed for the accomplishment of testing

   2. Test instructions and procedures that incorporate the requirements and acceptance criteria in applicable design documents

   3. Test prerequisites and the criteria for meeting these requirements and acceptance criteria

   4. Test items and the approach taken (including regression testing approaches) by the testing program

   5. Test logs, test data, and test results

   6. Test records that indicate the type of observation made, the results and acceptability, and the action taken in connection with any deficiencies or anomalies

   7. Test plans, test activities and task, test cases, and test coverage test methods and standards, and M&TE to support the tests

c. Verify that the results of testing are documented, reviewed, analyzed, and approved, by a qualified individual who was not responsible or involved in the design process to ensure test requirements have been fulfilled.

d. Verify that there is a documented method to identify and resolve discrepancies or anomalies between actual and expected integration and system validation test results.

e. Assess whether the process established to incorporate changes to the software or hardware or system configuration due to test results, is adequate to ensure that all test anomalies are documented and resolved.

NOTE: Anomalies discovered during testing may impact system, software, or hardware requirements. Verify that any modifications to the system, software, or hardware requirements to address the anomalies are adequately documented and configuration management activities for these modifications have been adequately performed.

f. Verify that DI&C system validation testing is conducted on a completely integrated system, in which all hardware and software functionality has successfully passed integration testing and have been combined into one final system. This includes interfaces with other systems (e.g., human-machine interfaces, displays, maintenance and testing equipment, and non-safety-systems).

03.06 Verification and Validation: Assess the independent verification and validation (IV&V) activities of the system development lifecycle by performing the following:

a. Verify that procedures are established and implemented for performing IV&V activities. The procedures should document the IV&V activities that need to be performed for each lifecycle phase.

b. Verify that the IV&V activities identified for each lifecycle phase has been adequately performed and documented.

c. Verify that the results of IV&V activities have been reviewed and approved by qualified individuals and appropriate management that are not within the line organization of those who perform the IV&V activities.

d. Verify that procedures are established for the documentation and resolution of all non-conformances identified during each life cycle phase.

03.07 Configuration Management Processes: Assess the configuration management activities of the system lifecycle by performing the following:

a. Verify that procedures are established and implemented for control of appropriate records for the system development activities:

- Identification and control of system requirements
- Identification and control of hardware and software requirements
- Identification and control of drawings and schematics (e.g., physical layouts, system configuration and interfaces, and functional logic diagrams)
- Documentation to support system, software, and hardware configurations

b. Verify that procedures are established and implemented for the control of appropriate records of software development activities which include:

- Identification and control of all software code
- Identification and control of all software design functional data (e.g., data templates and data bases)
- Identification and control of all software design interfaces
- Control of all software design changes
- Control of software documentation (e.g., user, operating, and maintenance documentation)

c. Verify that provisions are included in procedures to ensure software tools used to support system development and verification and validation processes are controlled under configuration management.

35710-04 RESOURCE ESTIMATE

Inspection resources necessary to complete this inspection procedure are estimated to be 160 hours of direct inspection per facility.

35710-05 REFERENCES

10 CFR Part 50, Appendix B, "Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants."

ANSI/ASME NQA-1, "Quality Assurance Program Requirements for Nuclear Facility Applications"

IMC 2502, “Construction inspection Program: Pre-Combined License (Pre-COL) Phase”

IMC 2507, “Vendor Inspections”

IMC 2508, “Construction Inspection Program: Design Certification.”

Regulatory Guide (RG) 1.152,"Criteria for Use of Computers in Safety Systems of Nuclear Power Plants." Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, 2023

RG 1.168, "Verification, Validation, Reviews and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants." Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission

RG 1.169. "Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants." Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission

RG 1.170, “Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants”

RG 1.171, “Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants”

RG 1.172, “Software Requirements Specifications for Digital Computer Software Used in Safety Systems of Nuclear Power Plants”

RG 1.173, “Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants”

Information Notice 86-77, “Computer Program Error Report Handling,” issued August 28, 1986

IEEE Std. 7-4.3.2-2016, "IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations"

IEEE Std. 603-1991, "IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations"

IEEE Std. 730-2002, “IEEE Standard Criteria for Software Quality Assurance Plans”

IEEE Std. 828-1990, “IEEE Standard for Configuration Management Plans”

IEEE Std. 829-1983, “IEEE Standard for Software Test Documentation”

IEEE Std. 830-1993, “IEEE Recommended Practice for Software Requirements Specifications”

IEEE Std. 1008-1987, “IEEE Standard for Software Unit Testing”

IEEE Std. 1012-1998, "IEEE Standard for Software Verification and Validation Plans

IEEE Std. 1028-1997, “IEEE Guide to Software Configuration Management”

IEEE Std. 1074-1995, "IEEE Standard for Developing Software Life Cycle Processes"

END

Attachment: Revision History for IP 35710

Attachment 1: Revision History for IP 35710

Commitment Tracking Number	Accession Number Issue Date Change Notice	Description of Change	Description of Training Required and Completion Date	Comment Resolution and Closed Feedback Form Accession Number (Pre-Decisional, Non-Public Information)
N/A	ML17278A510 01/30/18 CN 18-002	Initial issue to verify that safety-related software used for Digital Instrument and Control (DI&C) and Design and Analysis applications is developed in accordance with a Quality Assurance Program (QAP) that complies with the requirements of Appendix B to Title 10 of the *Code of Federal* Regulations (10 CFR) Part 50.	N/A	ML17278A511
	ML24022A098 04/18/24 CN 24-011	Title changed to reflect specific inspection requirements related to only Digital Instrument and Control (DI&C) software applications. Inspection requirements updated to include a more detailed DI&C software controls, including specific guidance for the software development process. Additional guidance added to differentiate between micro-processor based software and HDL-programmed devices. Detailed guidance updated to reflect inspection experience. Format updated to reflect current formatting guidance.	N/A	ML24036A289