ATTACHMENT 65001.22

INSPECTION OF DIGITAL INSTRUMENTATION AND CONTROL (DI&C) SYSTEM/SOFTWARE DESIGN ACCEPTANCE CRITERIA (DAC)-RELATED ITAAC

PROGRAM APPLICABILITY: 2503

65001.22-01 INSPECTION OBJECTIVES

01.01 To verify that the combined license (COL) holder (licensee) has developed the digital instrumentation and control (DI&C) system as committed in the licensing basis.

01.02 To confirm by inspection that the COL licensee has adequately implemented the DI&C development process to yield a system that meets the acceptance criteria in the Inspections, Tests, Analyses and Acceptance Criteria (ITAAC).

01.03 To provide implementation guidance for use of the Appendices.

65001.22-02 INSPECTION REQUIREMENTS AND GUIDANCE

02.01 Background. Inspection of ITAAC associated with a COL is intended to support the Commission finding stipulated in 10 CFR Part 52.103(g), specifically that the COL acceptance criteria (ITAAC acceptance criteria) are met, and that the facility has been designed and built to conform to the licensing basis. The Commission policy for Design Acceptance Criteria (DAC), as defined in SECY-92-053, allowed a licensee to provide implementation details for a DI&C design as ITAAC. The DI&C DAC-related ITAAC would be inspected as the development process for the systems progresses and the licensee completes the ITAAC throughout the facility post-COL (construction) phase.

02.02 Inspection Requirements and Guidance.

a. General Inspection Requirements. The development of safety-related DI&C systems and software should progress in accordance with a formally defined life cycle. Although life cycle activities may differ between licensees, all share certain characteristics. The staff's inspection and acceptance of digital safety system and software functions is based upon: 1) confirmation that acceptable plans were prepared to control software development activities; 2) evidence that the plans were implemented in the software development life cycles; and 3) evidence that the process produced acceptable design outputs.

Generic inspection attributes and criteria for each DI&C software life cycle phase are provided within Appendices 1 through 6 of this IP. It is recognized that not all DI&C life cycle phases may be inspected because they may not apply to each licensee's development program/process. The goal of this inspection activity is to examine the governing documents and samples of activities that demonstrate the implementation of these documents in order to provide a comprehensive inspection of the licensee's DI&C development process as deli-

neated in the ITAAC.

The actual planning and scheduling of the DI&C inspections is dependent on the licensee's design development schedule and associated milestones. The guidance contained herein is intended to mirror a typical development life cycle. Inspections should not be planned until the completion of life cycle phases by the licensee can be anticipated and expected completion dates can be confirmed. All construction inspection activities should be coordinated through the Region II Center for Construction Inspection (RII/CCI).

Specific Guidance. Gather pertinent information and discuss inspection planning and scheduling issues with the RII/CCI and/or Office of New Reactors (NRO) engineering technical experts. For example:

- importance/prioritization of activities
- concurrent inspections to be conducted using other IPs
- status and disposition of previous NRC findings
- licensee documented responses to applicable Generic Letters, Bulletins, Regulatory Issue Summaries and Information Notices
- commitments made in the COL pertaining to digital system/software development activities
- technical attributes that should be the focus of the inspection

Contact the licensee for information needed to prepare the inspection plan, for example:

- status of DI&C development activities, planned activities and schedule (used to focus inspection and determine inspection sample)
- identification of individuals assigned key positions and functions described by the licensee's Software Quality Assurance (QA) and Verification and validation (V&V) program
- availability of licensee personnel during the period tentatively scheduled for the inspection
- changes to Software QA or V&V program since any previous NRC inspection (e.g., policy, personnel, program description, implementing documents)

b. Requirements for Performance of Inspection. The inspection will be performed in accordance with the inspection plan. Adjustments to the inspection plan will be communicated to Region II/CCI to minimize impact to the licensee and to assist in revising inspection planning efforts accordingly.

Specific Guidance. Conduct the inspection in accordance with this IP and its associated appendices. The inspection should focus on safety-critical requirements of the digital I&C systems, including redundancy, independence between safety-related and non safety-related digital systems, independence of data communications, deterministic performance of trip and actuation functions, design simplicity (un-needed features implemented in safety systems), etc.

c. Requirements for Inspection Reporting. An inspection report and any findings will be prepared and approved in accordance with Inspection Manual Chapter 0613.

Specific Guidance. No specific guidance.

65001.22-03 RESOURCE ESTIMATE

The total estimated hours to complete this inspection, assuming all life cycle phases (all Appendices) are addressed, is 660 staff hours. A total of 80 hours each is allotted for inspection of Appendices 1, 5 and 6, and a total of 140 hours each is allotted for Appendices 2, 3 and 4. In addition, a total of 200 hours is estimated for preparation and documentation.

65001.22-04 REFERENCES

1. 10 CFR Part 52, “Licenses, Certifications, and Approvals for Nuclear Power Plants”
2. Regulatory Guide 1.206, C.II.1.2.5, “ITAAC for Instrumentation and Controls (SRP Section 14.3.5) and C.III.5, “Design Acceptance Criteria”
3. Regulatory Guide 1.152, Revision 2. "Criteria for Use of Computers in Safety Systems of Nuclear Power Plants." Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, 2006 (ML053070150)
4. Regulatory Guide 1.168, Revision 1. "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, 2004 (ML040410189)
5. Regulatory Guide 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, 1997 (ML003740102)
6. Regulatory Guide 1.170, “Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants”
7. Regulatory Guide 1.171, “Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants”
8. Regulatory Guide 1.172, “Software Requirements Specifications for Digital Computer Software Used in Safety Systems of Nuclear Power Plants”
9. Regulatory Guide 1.173, “Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants”
10. NUREG 0800 (SRP), Section 14.3, “Inspections, Tests, Analyses, and Acceptance Criteria”
11. NUREG 0800 (SRP), Branch Technical Position (BTP) 7-14, “Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems”
12. NUREG/CR-6101. "Software Reliability and Safety in Nuclear Reactor Protection Systems"
13. Inspection Manual Chapter 2503, “Construction Inspection Program: Inspections

of Inspections, Tests, Analyses, and Acceptance Criteria (ITAAC) Related Work”
14. Inspection Manual Chapter 0613, “Documenting 10 CFR Part 52 Construction and Test Inspections” (ML082490463)
15. ASME NQA-1, “Quality Assurance Requirements for Nuclear Facility Applications,” American Society for Mechanical Engineers
16. IEEE Std. 7-4.3.2-2003, "IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations"
17. IEEE Std. 603-1991, "IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations"
18. IEEE Std. 730-2002, “IEEE Standard Criteria for Software Quality Assurance Plans”
19. IEEE Std. 828-1990, “IEEE Standard for Configuration Management Plans”
20. IEEE Std. 829-1983, “IEEE Standard for Software Test Documentation”
21. IEEE Std. 830-1993, “IEEE Recommended Practice for Software Requirements Specifications”
22. IEEE Std. 1008-1987, “IEEE Standard for Software Unit Testing”
23. IEEE Std. 1012-1998, "IEEE Standard for Software Verification and Validation Plans"
24. IEEE Std. 1028-1997, “IEEE Guide to Software Configuration Management”
25. IEEE Std. 1074-1995, "IEEE Standard for Developing Software Life Cycle Processes"
26. IEEE Std. 1228-1994, "IEEE Standard for Software Safety Plans"
27. Inspection Procedure 65001.10, “Inspection of ITAAC-Related Installation of Instrument Components and Systems”
28. Inspection Procedure 52003, “Digital Instrumentation and Control Modification Inspection”

65001.22-05 PROCEDURE COMPLETION

Implementation of this IP is considered complete when the planned sample of attributes for the specified appendices is complete.

END

Appendices:

1. Inspection Guide for DI&C System/Software Life Cycle - Planning Phase
2. Inspection Guide for DI&C System/Software Life Cycle - Requirements Phase
3. Inspection Guide for DI&C System/Software Life Cycle - Design & Implementation Phase
4. Inspection Guide for DI&C System/Software Life Cycle - Integration Phase
5. Inspection Guide for DI&C System/Software Life Cycle - Validation & Test Phase
6. Inspection Guide for DI&C System/Software Life Cycle - Installation Phase

Attachment:

1. Revision History Sheet for IP 65000.22

Appendix 1 - Inspection Guide for DI&C System/Software Life Cycle - Planning Phase

A1.01 INSPECTION OBJECTIVES

Verify that the licensee's DI&C development process Planning Phase documents are consistent with the ITAAC design commitments and acceptance criteria.

A1.02 SAMPLE SIZE

Inspection of DI&C DAC-related ITAAC will typically rely on selection of a sample of attributes for verification. Given the importance of the various Life Cycle Plans in defining and detailing the quality design and development process expected for safety-related DI&C systems/software, inspection of a larger representative sample of attributes associated with each of the Planning Phase documents is appropriate. Initial sample size is left to the inspector's discretion based on review of inspection source documentation, and may be modified based on inspection issues and findings identified.

A1.03 INSPECTION REQUIREMENTS AND GUIDANCE

General Guidance.

A digital system/software development life cycle provides definition for a deliberate, disciplined, and quality development process. Implementation of this process should result in a quality DI&C system and supporting software. Verification of this process should confirm, by evaluation against applicable standards and criteria, that the licensee and vendor procedures and plans are sufficient to accomplish this goal.

The Planning Phase activities will provide documents that will be used to oversee the DI&C development project as it progresses from one Life Cycle Phase to the next. Compliance with RG 1.173 and IEEE-Std-1074 "Developing a Life Cycle Process," means mandatory activities are performed, requirements designated as "Shall" are met, and all inputs, outputs, activities and pre- and post-conditions mentioned by IEEE-Std-1074 are accounted for in the licensee's/applicant's life cycle model.

The documents resulting from the Planning Phase include the following minimum set; additional documents may be required by the development organization as part of their standard business procedures. It should be noted that software life cycle Plans for Operations, Maintenance and Training are not included; these elements are not considered part of the digital I&C DAC envelope, and can be covered through DI&C system as-built inspection.

- Software Management Plan (SMP)
- Software Quality Assurance Plan (SQAP)
- Software Configuration Management Plan (SCMP)
- Software Verification and Validation Plan (SVVP)

- Software Safety Plan (SSP)
- Software Development Plan (SDP)
- Software Integration Plan (SIntP)
- Software Installation Plan (SInstP)
- Software Test Plan (STP)

Generally, these Planning documents include management characteristics, implementation characteristics, and resource characteristics. Not all specific characteristics occur for every Plan. Management characteristics for each Plan shall include a stated Purpose, identify Organizational and Oversight responsibilities, and account for risk and security management. Implementation characteristics shall include Process Metrics as well as guidance on Procedure Control and Recordkeeping. Resource characteristics shall include details of Special Tools utilized in the development process, Personnel resources and qualification, and the Standards used to meet regulatory requirements. Inspection should focus on those aspects of the Plans which can impact the safety and quality of the resulting DI&C system/software.

The inspectable attributes identified in the following sections were compiled from many of the references listed in this procedure. Additionally, other attributes may be identified in the Acceptance Criteria of the specific ITAAC. These additional attributes should be included in the scope of the Plan inspection. This inspection procedure verifies commitments made in the COL and licensing basis.

Inspection Requirements.

A1.03.01 Inspection of Software Management Plan (SMP)

a. Verify that the SMP addresses the following specific management aspects of the software development project, as committed to in the licensing basis:

    1. Organizational structure is defined. Responsibilities are known and documented, and a management structure exists to keep the SMP up to date through a configuration control process.
    2. Oversight of vendors. The SMP should describe the interaction between licensee and system/software vendors, extension of QA requirements to vendors, what checks and audits the licensee will perform and their impact.
    3. Independence between the software development group and the QA group, system/software safety group, and V&V group. If independence aspects are described in the planning documents of these organizations, such as the V&V Plan, Safety Plan or QA plan, the SMP should provide a pointer to those plans.
    4. Personnel responsible for various items have the experience, training and qualifications to perform those duties.

b. Verify that the SMP includes the following key attributes, as committed to in the licensing basis:

1. Project schedule includes time allotted for review (management, V&V, etc.) and audit.
2. Project work products and deliverables are adequately defined.
3. Responsibilities documented and communicated to the development organization.
4. Project constraints that may have an impact on safety are identified.
5. Known risk factors identified.
6. Required reports and technical documents identified.
7. Training requirements known and documented.
8. Internal review and audit processes identified.

A1.03.02 Inspection of Software Quality Assurance Plan (SQAP)

a. Many aspects of software quality are described in the various life cycle Plans. These include the Configuration Management Plan, the Software Safety Plan, and the Software Verification and Validation Plan.

   The SQAP shall comply with the requirements of 10 CFR Part 50, Appendix B, and the licensee's approved QA program. The SQAP should typically: 1) identify which QA procedures are applicable to specific software processes; 2) identify particular methods chosen to implement QA procedural requirements; and 3) augment and supplement the QA program as needed for software.

   Verify that the SQAP addresses the following, as committed to in the licensing basis:

   1. Management Tasks
   2. Documentation
   3. Recordkeeping
   4. Standards, Practices, Conventions
   5. Reviews and Audits
   6. Problem Reporting and Corrective Action
   7. Control of Tools, Techniques, and Methodologies
   8. Supplier (Vendor) Control
   9. Version Control
   10. Audit Trails

b. Verify that the SQAP includes the following key attributes, as committed to in the licensing basis:

   1. SQAP specifies which software products are covered by the Plan.
   2. Project elements (organizations) that interact with the QA organization are listed.
   3. Organization engaged in software QA activities is independent of the development organization, including cost and schedule.
   4. Life Cycle development phases that will be subject to QA oversight are listed.
   5. Required QA tasks are listed and described.
   6. Conflict resolution among organizations is described.

7. Required software documents are listed.
8. Required reviews and audits are listed.
9. Methods by which each review and audit will be carried out is described.
10. SQAP includes provisions to assure that problems will be documented and corrected.

A1.03.03 Inspection of Software Configuration Management Plan (SCMP)

a. Verify that the SCMP addresses the following specific activities, as committed to in the licensing basis:

1. Production/development baselines are identified and established.
2. Review, approval, and control of changes is defined.
3. Tracking and reporting of changes is defined
4. Audits and reviews of the evolving products are established.
5. Control of interface documentation is defined.

b. Verify that the SCMP includes the following key attributes, as committed to in the licensing basis:

1. Product interfaces that have to be supported within the project are identified.
2. The required capabilities of the staff needed to perform SCM activities are defined.
3. The responsibilities for processing baseline changes are defined.
4. The SCMP specifies who is responsible for each SCM activity.
5. The organizational interfaces that affect the SCM process are identified.
6. SCM activities that will be coordinated with other project activities are described.
7. Describes how phase-specific SCM activities will be managed during the different life cycle phases.
8. Specific procedures exist to manage the change process.
9. Audit procedures are defined.
10. Configuration identification scheme matches the structure of the software product.
11. SCMP specifies which items will be placed under configuration control (configuration items (CI)).
12. SCMP describes the authority of the Configuration Control Board (CCB).
13. CCB authority is sufficient to control safety-related changes to the CI baseline.
14. SCMP requires the CCB to assess the safety impact of change requests.
15. Provisions are included for auditing the SCM process.
16. SCMP provides for periodic reviews and audits of the configuration baseline, including physical audits of the baseline.
17. SCMP provides for audits of suppliers and subcontractors, if such are used.
18. SCMP accounts for all assets, including backup and recovery software.

A1.03.04 Inspection of Software Verification & Validation Plan (SVVP)

a. Verify that the SVVP addresses the following specific activities, as committed to in the licensing basis:

1. Management of Life Cycle V&V. The major portion of the V&V Plan will describe the methods in which V&V will be carried out through the life of the development project. In general, the following activities should be required for each phase of the life cycle:

   (a) Identify the V&V tasks for the life cycle phase.
   (b) Identify the methods that will be used to perform each task.
   (c) Specify the source and form for each input item required for each task.
   (d) Specify the purpose, target and form for each output item required for each task.
   (e) Specify the schedule for each V&V task.
   (f) Identify the resources required for each task.
   (g) Identify the risks and assumptions associated with each V&V task.
   (h) Identify the organizations or individuals responsible for performing each V&V task.

2. Requirements Phase V&V. The V&V Plan should describe how the various V&V tasks will be carried out for the following:
   (a) Software Requirements Traceability Analysis
   (b) Software Requirements Evaluation (Report)
   (c) Software Requirements Interface Analysis
   (d) System Test Plan Generation
   (e) Acceptance Test Plan Generation

3. Design & Implementation Phase V&V. The V&V Plan should describe how the various V&V tasks will be carried out for the following:
   (a) Software Design Traceability Analysis
   (b) Software Design Evaluation (Report)
   (c) Software Design Interface Analysis
   (d) Test Plan Generation
   (e) Source Code Traceability Analysis
   (f) Source Code Evaluation
   (g) Source Code Interface Analysis
   (h) Source Code Documentation Analysis

4. Integration Phase V&V. The V&V Plan should describe how the various V&V tasks will be carried out for the following:
   (a) Integration Test Procedure Generation
   (b) Integration Test Procedure Execution

5. Validation & Test Phase V&V. The V&V Plan should describe how the various V&V tasks will be carried out for the following:
   (a) Acceptance Test Procedure Generation
   (b) System Test Procedure Execution
   (c) Acceptance Test Procedure Execution

6. Installation Phase V&V. The V&V Plan should describe how the various V&V tasks will be carried out for the following:
   (a) Installation Configuration Audit
   (b) Final V&V Report Generation

b. Verify that the SVVP includes the following key attributes, as committed to in the licensing basis:

1. SVVP references the SMP and/or SQAP.
2. Specific elements of the higher-level plans are addressed in the SVVP.
3. Scope of the V&V effort is defined.
4. SVVP describes the V&V organization, including its relationship to the development organization (must be independent).
5. Schedule is defined that provides enough time for V&V activities to be carried out. The V&V organization shall have freedom from scheduling constraints or impact.
6. Tools, techniques, and methods to be used in the V&V process defined. SVVP identifies method of handling anomalies encountered during each activity.
7. V&V schedule and resource requirements are described in detail.
8. SVVP identifies the responsibilities, line organization and reporting requirements for the V&V organization.
9. SVVP defines procedure for management review of the V&V process.
10. Process is developed for periodic assessment and updating of V&V procedures and tools.
11. Defined procedure is developed for correlating V&V results with management and technical review documents.
12. SVVP is coordinated with project planning documents to ensure early availability of the planning documents for the V&V effort.
13. SVVP explicitly defines and describes the following (as a minimum) V&V tasks:
    (a) Audits
    (b) Regression testing and analysis
    (c) Security assessment
    (d) Testing evaluation
    (e) Documentation evaluation

c. Verify that the SVVP includes the following life cycle phase-specific attributes, as committed to in the licensing basis:

1. Requirements Activities
   (a) Concept documentation, software requirements specification (SRS), interface requirements, hazards analysis, and user documentation will be complete prior to beginning the V&V requirements analysis.
   (b) SVVP explicitly defines the activities required during the requirements analysis.
   (c) SVVP requires the performance of a software requirements traceability analysis.
   (d) SVVP requires that the SRS be evaluated for performance issues.

(e) SVVP requires that a system test plan and an acceptance test plan be generated during the requirements phase.

2. Design & Implementation Activities
   (a) SVVP requires the generation and dissemination of anomaly reports.
   (b) SVVP explicitly defines the activities required during the design and implementation phase.
   (c) SVVP requires the performance of a design traceability analysis that traces elements of the detailed design and coding to elements of the software requirements.
   (d) SVVP requires a design evaluation (report).
   (e) SVVP requires a design interface analysis.
   (f) SVVP requires that the software design document be evaluated against hardware requirements, operator requirements, and software interface requirements documentation.
   (g) SVVP requires a software component test plan, an integration test plan, and a test design be generated for use in later testing.
   (h) SVVP requires that the source code be evaluated for correctness, consistency, completeness, accuracy, readability, safety, and testability.
   (i) SVVP requires generation and use of test cases to help ensure the adequacy of test coverage.
   (j) SVVP requires the generation of test cases for software component, integration, system, and acceptance testing.

3. Integration, Validation & Test, and Installation Activities

   (a) SVVP explicitly defines the activities required during the integration and validation analysis and testing.
   (b) SVVP requires the performance of sufficiently detailed testing requirements so as to ensure that there is a very low probability of error during operation, and describes the consequences if errors were to occur.
   (c) SVVP explicitly defines the activities required during the installation analysis and testing.
   (d) SVVP requires the performance of an installation configuration audit.
   (e) SVVP requires the generation of a final report.

A1.03.05 Inspection of Software Safety Plan (SSP)

a. Verify, consistent with commitments in the licensing basis, that the SSP addresses the following documentation that will be required as part of the software safety program:
   1. Results of all safety analyses
   2. Information on suspected or verified safety problems
   3. Results of audits performed on software safety program activity
   4. Results of safety tests carried out on the software system

5. Records on training provided to software safety personnel and software development personnel

b. Verify that the SSP includes the following key attributes, as committed to in the licensing basis:

1. Software safety organization is described and authority defined; authority sufficient to enforce compliance with safety requirements and practices.
2. SSP provides a mechanism for defining safety requirements, performing software safety analysis tasks, and testing safety-critical features of the DI&C system.
3. SSP describes what safety-related documents will be produced during the development life cycle; contents sufficient to ensure that known safety concerns are addressed in the appropriate places within the development life cycle.
4. SSP identifies the safety-related records that will be generated, maintained, and preserved.
5. SSP specifies the process of approving and controlling software tool use.
6. SSP provides a means to ensure that safety-critical software developed by a subcontractor meets the requirements of the software safety program.

A1.03.06 Inspection of Software Development Plan (SDP)

a. Verify, consistent with commitments in the licensing basis, the following:
1. SDP defines the tasks that are a part of each life cycle phase.
2. SDP defines life cycle phase inputs and outputs, including review, verification and validation of those outputs.
3. SDP lists the international, national, industry, and company standards and guidelines, including regulatory guides, which will be followed, and whether or not these standards and guidelines have previously been approved by the NRC staff.

b. Verify that the SDP includes the following key attributes, as committed to in the licensing basis:

1. Technical standards that will be followed are listed.
2. Technical milestones are listed.
3. Milestones are consistent with the schedule provided in the SMP.
4. Technical documents that must be produced are listed.
5. Technical documents are consistent with those listed in the SMP.
6. Milestones, baselines, reviews, and signoffs are listed for each document.
7. Audit reports document that the SDP is being followed.

A1.03.07 Inspection of Software Integration Plan (SIntP)

a. Verify, consistent with commitments in the licensing basis, the following:
1. SIntP describes the general strategy for integrating the software modules together into one or more programs.
2. SIntP integration strategy includes integrating the various software mod-

ules together to form single programs.
3. SIntP integration strategy includes integrating the software with the hardware and instrumentation, and testing the resulting integrated product.
4. SIntP integration strategy includes regression testing.

b. Verify that the SIntP includes the following key attributes, as committed to in the licensing basis:

1. SIntP specifies the levels of integration required.
2. SIntP is consistent with the software design specification.
3. SIntP describes each step of the integration process.
4. SIntP describes the environment that will be used to perform and test each integration step.
5. Software and hardware tools that will be used to integrate the computer system are listed.
6. SIntP includes instructions on how to carry out integration steps.
7. SIntP includes a contingency plan in case the integration fails.
8. SIntP includes a requirement for configuration control of the completed product.

A1.03.08 Inspection of Software Installation Plan (SInstP)

a. Verify that the SInstP includes the following key attributes, as committed to in the licensing basis:
1. General procedures for installing the software product are described.
2. Materials required are listed in an Installation Package.
3. Complete step-by-step procedures exist for installation in the operational environment.
4. Expected results from each installation step are described.
5. Known installation error conditions and recovery procedures are described.
6. Installation Plan is fully tested.

b. Verify that the SInstP exhibits the following characteristics, as committed to in the licensing basis:

1. Includes a description of the System operating environment.
2. Includes a description of the organization responsible for installation, their responsibilities, and interfaces with other organizations.
3. Includes acceptance criteria to determine success/failure of the installation effort.
4. Includes a description of procedures for System, combined hardware/software, and software installation.
5. Includes requirements for functional checks of the installation, including checks for correct version of the application software and source code on the correct hardware (platform).
6. Includes requirement for adequate safety function testing.
7. Includes requirement that installation tools be qualified to a degree commensurate with the safety significance of the installed software.

A1.03.09 Inspection of Software Test Plan (STP)

a. **Note** – Attributes of the STP may also be verified as part of the Validation and test Phase activities in Appendix 5. Verify that the STP includes the following key attributes, as committed to in the licensing basis:

    1. General testing for the software, including unit, integration, factory and site acceptance, and installation testing is described.
    2. Testing responsibilities have been assigned to appropriate personnel groups (test group and V&V group).
    3. Provisions are included for re-test in the event of test failures.
    4. Provisions are included for full testing after any software modifications.
    5. Provisions are included for the V&V group to have oversight of final System (hardware and software) testing.

b. Verify that the STP exhibits the following characteristics, as committed to in the licensing basis:

    1. Testing organization and interfaces are described.
    2. Scope of testing activities is described.
    3. Test scope includes secure development and operational environment (SDOE) testing strategy.
    4. Acceptance criteria are included to determine the success or failure of testing activities.
    5. Procedures for testing each software item are identified.
    6. Methods for synchronizing procedures and test cases with the software design are described.
    7. Procedures for tracking problem reports and their resolution are described.
    8. Software test record keeping requirements are described.

Appendix 2 - Inspection Guide for DI&C System/Software Life Cycle - Requirements Phase

A2.01 INSPECTION OBJECTIVES

Verify that the licensee's DI&C development Requirements Phase process and documentation are consistent with the ITAAC design commitments and acceptance criteria.

A2.02 SAMPLE SIZE

Inspection of DI&C DAC-related ITAAC will typically rely on selection of a sample of attributes for verification. Inspection of Requirements Phase activities will be accomplished through verification of selected attributes for a representative sample of critical system/software requirements (typically 12 - 15 critical requirements per inspector). Traceability (through thread audit techniques) should be verified for the same representative sample of critical requirements. Sample size should be expanded at the inspector's discretion if inspection issues or findings are identified.

A2.03 INSPECTION REQUIREMENTS AND GUIDANCE

General Guidance.

The scope and activities associated with the Requirements Phase result in a complete description of what the Digital I&C System (hardware and software) must accomplish, and establishes functional requirements for the system software. **Note** - any reference to the Digital I&C System implies both the hardware and software elements. Activities include:

- Translation of functional and regulatory requirements to DI&C system(s) requirements
- Define and document I&C System (hardware and software) requirements
- Define, document, prioritize, and integrate Software requirements
- Define and document Software Interface and Performance Requirements
- Requirements Safety Analysis
- Requirements Verification

The product of these activities will be documented in a Software Requirements Specification (SRS). Inspection will verify that System and Software Requirements were developed in accordance with applicable codes, standards and regulations. Inspection should focus on critical requirements essential for robust I&C system (hardware and software) design, including Redundancy, Independence (both physical and data communication), Deterministic performance, and design Simplicity. The SRS is an essential part of the record of the design of safety-related system software, and is a design input to the remainder of the software development process. Software

requirements serve as the design bases for the software to be developed.

Inspection Requirements.

A2.03.01 Digital I&C (DI&C) System Requirements

a. DI&C System requirements will form the basis for the DI&C System design specification. Verify the following attributes for the DI&C system, as committed to in the licensing basis:

    1. Each DI&C System requirement is separately identified.
    2. DI&C System requirements satisfy applicable codes, standards and regulatory requirements.
    3. DI&C System requirements are correctly derived from originating or source (facility) requirements (for functionality and performance) and safety requirements (from the facility Safety Analysis).
    4. Critical DI&C System safety considerations are identified.
    5. Critical DI&C System physical and SDOE considerations are identified.
    6. Critical DI&C System boundary value testing requirements are identified.
    7. DI&C System Source Requirements Lists (or equivalent index) are developed and requirements traceability defined/documented.

        Note - a Requirements Traceability Matrix (or equivalent tracking/traceability tool) should show, as a minimum, each requirement, the source of the requirement, the life cycle phases that are utilized by the development project, the testing to be applied to each requirement, and the associated requirement item identification. This will allow the matrix to be reviewed in order to assure that each requirement is addressed by the output products of each phase. The matrix will allow life cycle phase-to-phase and end-to-end review. The inspector should be able to trace requirements through the development life cycle, forward and backward. The traceability matrix can also be used to assist in impact assessment as requirements change, and provides documentation that safety requirements and licensing commitments are met.

b. Verify traceability of functional, regulatory and design basis (source) requirements to DI&C system requirements.

    1. Using the Traceability Matrix (or equivalent traceability tool), select a representative sample of DI&C system requirements and trace them backwards to the source requirements from which they were derived.
    2. Using the Traceability Matrix (or equivalent traceability tool), select a sample of source requirements and trace them forward to verify that the DI&C system requirements were correctly derived. Where possible, select source requirements that provide more than one system requirement (output), and trace the outputs accordingly.

3. Verify that forward traceability also includes an output to a test and validation process. Test and validation will ultimately confirm that requirements have been met.
   Note - tracing activities are designed to identify any source requirements not adequately addressed by the System requirements, System requirements not meeting the intent of source requirements, and any missing System requirements.

A2.03.02 Software Requirements

a. The DI&C software requirements are derived from the DI&C system requirements. Verify that software requirements exhibit the following attributes, as committed to in the licensing basis:

   1. Software functional requirements are individually identified.
   2. Software functional requirements are unambiguously stated.
   3. Software functional requirements specify what the software must do.
   4. Each software functional requirement is verifiable through inspection or testing of the completed software product (includes application software and source code).
   5. Software requirements are defined in a Software Requirements Specification (SRS).

b. Verify that the SRS lists and defines the following software interface requirements, as committed to in the licensing basis:

   1. Interfaces between software systems (software-to-software) are defined.
   2. External interfaces to software (user, hardware) are specified.
   3. Communication protocols are defined, including error detection methods.
   4. Input/Output (I/O), with validation methods of sensors/actuators is adequately described.
   5. Instrumentation parameters and values are adequately described, and conversion algorithms are defined.

c. Verify that the SRS lists and defines the following software performance attributes, as committed to in the licensing basis:

   1. Static performance requirements (number of terminals, simultaneous users, etc.) are adequately described.
   2. Timing requirements are described and specified numerically.
   3. All performance requirements are individually identified.
   4. Each performance requirement is testable.

d. Verify traceability of DI&C system requirements to software requirements.

   1. Using the Traceability Matrix (or equivalent traceability tool), select a representative sample of critical software requirements and trace them backwards to the DI&C System requirements from which they were derived.

2. Using the Traceability Matrix (or equivalent traceability tool), select a sample of DI&C System requirements and trace them forward to verify that the software requirements were correctly derived. Where possible, select system requirements that provide more than one software requirement (output), and trace the outputs accordingly.
3. Verify that the selected software requirements are included in software test documentation.

   Note - each identifiable requirement in an SRS must be traceable backwards to the system requirements and the design bases or regulatory requirements that it satisfies. Each identifiable requirement should be written so that it is also "forward traceable" to subsequent design outputs, e.g., from SRS to design, implementation, integration, and validation stages of the development project. Forward traceability to all documents resulting from the SRS includes verification and validation materials. For example, a forward trace should exist from each requirement in the SRS to the specific test used to confirm that the requirement has been met.

   Tracing activities are designed to identify any system requirements not adequately addressed by the software requirements, software requirements not meeting the intent of system requirements, and any missing software requirements.

A2.03.03 Requirements Phase Documentation

a. Verify that development processes supporting Requirements Phase activities are documented, as committed to in the licensing basis. Verify the following:
   1. Audit requirements are specified.
   2. System and Software Requirements Review process is defined and implemented.
   3. Configuration Management process (change control, version control, audit trails, etc.) process is defined and implemented.
   4. Independent Verification and Validation (IV&V) process is defined and implemented.

b. Verify the following documentation exists for the Requirements Phase activities, as committed to in the licensing basis:

   1. Audit Report and any Condition Reports for issues, nonconformances and audit findings.
   2. Documentation of corrective action stemming from Condition Reports.
   3. Requirements Configuration Management Report; verify that configuration change orders are communicated to the IV&V and Test organizations.
   4. Requirements Verification (IV&V) Report; verify that the report confirms that there is 100% verification of software requirements by the IV&V organization.

A2.03.04 Requirements Safety Analysis

a. A Safety Analysis encompassing Requirements Phase activities should be performed and documented. The analysis should determine which software requirements are critical to system safety, that all safety requirements imposed by the DI&C System design have been correctly addressed in the SRS, and that no additional hazards have been created by the SRS. Verify the following attributes and results of the Safety Analysis, as committed to in the licensing basis:

    1. DI&C system safety requirements are correctly included in the SRS.
    2. Each DI&C system safety requirement can be traced to one or more software requirements.
    3. Each software requirement can be traced to one or more system requirements.
    4. There are no missing or inconsistently specified DI&C system functions or software requirements (from the selected sample).
    5. No new hazards have been introduced to the process.
    6. Requirements that can affect safety have been identified.
    7. Safety issues and resolutions are documented.

b. Analyses of software safety requirements should be performed to determine the impact of incorrectly developed or incorporated requirements on safety-related software. Verify that the following elements have been considered:

    1. Diverse Requirements for the DI&C system and software are included based on a Diversity and Defense-in-Depth (D3) analysis.
    2. Critical/Non-Critical Software Requirement discriminators and criteria, based on criticality analysis that determines the impact of a failure to meet software requirements, are included.
    3. Requirements Traceability, as a mechanism to determine compliance with DI&C system specifications, is included.
    4. Specific Requirements are included to ensure deterministic behavior of the DI&C system.
    5. Specific Requirements are included to address Secure Development and Operational Environment (per RG 1.152) vulnerabilities.

Appendix 3 - Inspection Guide for System/Software Life Cycle - Design & Implementation Phase

A3.01 INSPECTION OBJECTIVES

Verify that the licensee's DI&C development Design and Implementation Phase process and documentation are consistent with the ITAAC design commitments and acceptance criteria.

A3.02 SAMPLE SIZE

Inspection of DI&C DAC-related ITAAC will typically rely on selection of a sample of attributes for verification. Inspection of Design Phase activities will be accomplished through verification of attributes for a representative sample of critical design elements (typically 12 - 15 design elements per inspector). Traceability (through thread audit techniques) should be verified for the same representative sample of critical design elements. Sample size should be expanded at the inspector's discretion if inspection issues or findings are identified.

A3.03 INSPECTION REQUIREMENTS AND GUIDANCE

General Guidance.

The follow-on to development of software requirements is the design of the software system. Activities include the documentation, analysis, and review of the various software design processes. Design control products/documents (Design Outputs) resulting from these activities includes:

- Hardware and Software Architecture Descriptions
- Software Design Specification
- Code Listings

The software design activities translate the software requirements specifications into a hardware/software architecture specification and a software design specification. Reviewing the software architecture of the digital I&C system allows the staff to understand how the high-level coded functions of the system interact to accomplish the design function. Review of the Software Design Specification enables the staff to ensure that the software code accurately reflects the software requirements.

Inspection Requirements.

A3.03.01 System Architecture

a. Verify that a Hardware/Software Architecture Description (or equivalent document) has been prepared, as committed to in the licensing basis. The Architecture Description should be correct, consistent, unambiguous, and verifiable. All major hardware and all major software processes should be included.

Verify that the Architecture Description includes mapping of software to hardware, mapping of logical and physical communication paths, and descriptors for independent software elements.

b. Verify that the Architecture Description addresses the following attributes, as committed to in the licensing basis:

1. Design architecture shows how the various hardware elements are connected together.
2. Independent software elements are shown in the design architecture. This includes:
   - Processes, which perform computations
   - Files and databases, which store information
   - I/O messages, which receive and transmit information
   - Displays and human/machine interface (HMI)
   - Communication, which moves information among processes, files and databases, I/O channels, and HMI
3. Design architecture shows how the various software elements are logically connected together
4. Design architecture shows how each software element is mapped to a hardware element

A3.03.02 Software Design Specification (SDS)

a. The SDS should depict exactly how the software requirements will be implemented in software modules and programs. It should be correct, complete, internally consistent, unambiguous, verifiable, and testable. Each Design Element in the SDS should be traceable to one or more specific software requirements.

Verify that each Design Element exhibits the following attributes:

- Name of the Design Element
- Type (system, subsystem, module, database, file, data structure, screen display, message, program, or process)
- Purpose/Function of the Design Element (why the element exists in the design)
- Detailed description of the ways in which the Design Element interacts with other design elements.

A list of all requirements implemented by the Design Element may also be included. This is used to verify that the design will implement the requirements, and only the requirements. Some elements may implement more than one requirement, while some requirements may need several elements for a successful implementation.

b. Select a representative sample of Design Elements from the SDS. Verify that the Design Elements address the following attributes, as committed to in the

licensing basis:

1. Every software requirement listed in the Software Requirement Specification (SRS) can be traced to one or more specific design elements that implement the requirement.
2. Every design element can be traced to one or more specific software requirements that the design element implements.
3. Documentation exists to demonstrate that there are no unintended functions in the design.
4. Static and dynamic structures exhibit minimal connections between design elements.
5. Safety-critical functions are separated from normal operating functions, with well-defined interfaces between them.
6. Design Element - if any of the following concepts are used in the design, adequate justification is given for their use:
   - Point arithmetic
   - Interrupts, except for periodic timer interrupts
   - Multi-processing on a single processor
   - Dynamic memory management
   - Event-driven communications between processes
   - Time-critical subroutines
7. Inputs to each software module are checked for validity.
8. Auto swap-over to backup software is built into the design.
9. For commercial-off-the-shelf (COTS) software that is part of the design, non-required features can be disabled.

c. Verify traceability of software requirements to SDS design elements.

1. Using the Traceability Matrix (or equivalent tool), select a representative sample of software Design Elements and trace them backwards to the software requirements from which they were derived.
2. Using the Traceability Matrix (or equivalent tool), select a representative sample of Design Elements and trace them forward to verify they were properly coded in the detailed design.
3. Trace the selected software design element to the test(s) that will be used to confirm that the element has been correctly designed.

   Note - tracing activities are designed to identify any software requirements not adequately addressed by the design elements, design elements not meeting the intent of software requirements, and any missing design elements.

A3.03.03 Detailed Design & Implementation (Code Development)

a. Implementation consists of the translation of the completed software design into code and data stores. The risks involved in writing the code are that the design may not be correctly implemented in the code, or coding errors may add additional hazards. Verify the following, as committed to in the licensing basis:

1. Application software and system code is compiled in a Code Listing.
2. All system code is documented.

b. Verify that, as the Code Listing was developed, the following Code functional characteristics were considered, as committed to in the licensing basis:

1. Accuracy requirements were considered.
2. The System is coded such that corrupted data will not have an adverse impact on the system.
3. No new hazards or security threats are introduced by the Code.
4. The Code is written such that system execution and timing is deterministic.

A3.03.04 Design Phase Documentation

a. Verify that development processes supporting Design Phase activities are documented, as committed to in the licensing basis. Verify the following:

1. Audit requirements are specified.
2. Design Review process is defined and implemented.
3. Configuration Management (change control) process is defined and implemented.
4. Independent Verification and Validation (IV&V) process is defined and implemented.

b. Verify the following documentation exists for the Design Phase activities, as committed to in the licensing basis:

1. Audit report and any Condition Reports for issues, nonconformances, and audit findings.
2. Documentation of corrective actions stemming from Condition Reports.
3. Design Configuration Management Report; verify that configuration change orders are communicated to the IV&V and Test organizations.
4. Design Verification (IV&V) Report; verify that the report confirms 100% verification of design elements and correlation to code elements by the IV&V organization.

A3.03.05 Design Safety Analysis

a. The Design Safety Analysis verifies that the design correctly and consistently incorporates the system safety requirements, identifies safety-critical software design elements, and detects errors that might result in violations of the system safety requirements. The analysis should ensure that all safety critical requirements have been included in the design, and that no new design features are developed that have no basis in the requirements. The analysis should identify any conditions that would prevent safety-critical processing from being accomplished. Select a sample from the following attributes and verify that they have

been included in the Safety Analysis, as committed to in the licensing basis:

1. Logic

   - Equations and algorithms in the software design correctly implement safety-critical requirements.
   - Control logic on the software design completely and correctly implements the safety-critical requirements.
   - Control logic correctly implements error handling, off-normal processing, and emergency processing requirements. The system must be recoverable (e.g. from a power loss)
   - Design logic is such that design elements that are not considered safety-critical cannot adversely affect the operation of the safety-critical design elements. The system must be "fail-safe."

2. Data

   - Safety-critical data items are identified by type, unit, range, and error bounds.
   - Analog inputs (from field devices) are properly scaled.
   - Design elements that can change a safety-critical data item are known.
   - No interrupt will change the value of a safety-critical data item in an unanticipated manner.

3. Interface

   - Control linkages between design elements are correctly and consistently designed.
   - All parameters passed between design elements are consistent in type, structure, physical units, and direction (input/output).
   - No safety-critical data item is used before being initialized.

4. Constraints

   - Design constraints listed in the SRS have been followed in the design.
   - Known external limitations on the design have been recognized and included in the design (includes hardware limitations, instrumentation limitations, operation of the system equipment, etc.).
   - Design meets timing and sizing requirements.
   - Equations and algorithms work across the complete range of input data item values.
   - Equations and algorithms provide sufficient accuracy and response times as specified in the SRS.

b. A Code Safety Analysis should be performed and documented. The analysis should determine that the code correctly implements the software design, does not violate any safety requirements and that no additional hazards have been created by the coding activity. Select a sample from the following attributes and verify that they have been included in the Code Safety Analysis, as committed to in the licensing basis:

1. Logic

- Code logic correctly implements the safety-critical design criteria.
- Design equations and algorithms are correctly implemented in the code.
- Error handling design is correctly implemented in the code.
- Off-normal and emergency operations design is correctly implemented in the code.
- Non-critical code cannot adversely impact the function, timing, and reliability of safety-critical code.
- Interrupts included in the code will not take precedence over, or prevent the execution of, safety-critical code modules.
- Software failures and compensatory measures are implemented.
- Overrides and bypasses are implemented.

2. Data

- Definition and use of data items in the code are consistent with the software design.
- No safety-critical data item can have its value changed in an unanticipated manner, or by an unanticipated module.
- No interrupt can destroy safety-critical data items.

3. Interface

- Parameters passed between code modules are analyzed for consistency, including typing, structure, physical units, and number and order of parameters.
- Direction of parameters is consistent, both internally in the code, and externally with the software design.
- External interfaces are evaluated for correct format of messages, content, timing, and consistency.

4. Constraints

- Adequate memory space is allocated for the safety-critical code and data structure. Consider normal, off-normal, and emergency operating modes.
- Actual timing of events in the code is consistent with the timing analysis performed as part of the software design.

- All timing requirements are met.

c. Failure Analysis

A Failure Modes and Effects Analysis (FMEA) or other failure analysis should be performed to ensure that single failure requirements associated with system safety analyses and assumptions are confirmed. The FMEA should include system architecture to ensure that key design principles of redundancy and independence have been incorporated and that single failure requirements are met (Refer to NRC RG 1.53 “Single Failure Criterion). The FMEA attributes are incorporated from IEEE-Std-352 “Reliability Analysis.” Verify the following from the FMEA or equivalent failure analysis:

1. Credible failure modes for the system/software were identified.
2. Impact (failure effect) on the system/software was evaluated.
3. Provisions to compensate for failure(s) are identified and included.

Appendix 4 - Inspection Guide for System/Software Life Cycle - Integration Phase

A4.01 INSPECTION OBJECTIVES

Verify that the licensee's DI&C development Integration Phase process and documentation are consistent with the ITAAC design commitments and acceptance criteria.

A4.02 SAMPLE SIZE

Inspection of DI&C DAC-related ITAAC will typically rely on selection of a sample of attributes for verification. Inspection of Integration Phase activities will be accomplished through verification of attributes for a representative sample of integrated builds (typically 12 - 15 builds per inspector). Additionally, traceability should be conducted for the same sample of integrated builds (software subsystem or software system) backward to the code elements contained in the build, and forward to the software field installation. Sample size should be expanded at the inspector's discretion if inspection issues or findings are identified.

A4.03 INSPECTION REQUIREMENTS AND GUIDANCE

General Guidance.

Integration consists of the activities that combine the various software and hardware components into a single system. Software integration actually consists of three major phases:

- Integrating the various software modules together to form single executable programs
- Integrating the single programs with the hardware and instrumentation
- Testing the resulting integrated product

The Software Requirements and the Software Detailed Design should be analyzed to determine the order for combining software components into an overall system. The Integration Plan should include the tools, techniques, and methodologies needed to perform the integration.
System Build Documents (SBDs) are generally needed to verify that the programs actually delivered and installed are the programs that underwent the V&V process and were tested. Any future maintenance, modifications or updates will require that the maintainers know which version of the programming to modify. As a result, the System Build Documents are closely tied to the configuration management program.

Inspection Requirements.

A4.03.01 System/Software Integration

a. SBDs should be developed. The SBDs describe precisely how the system hardware and software components are combined into an operational system. The SBDs are a Design Output of the Integration Activities. From a configuration control standpoint, the items included in the SBD should be sufficient to show that the programming listed in the build documentation is identified by version, revision, and date, and that the version and revision was tested and found appropriate. Verify the following, as committed to in the licensing basis:

1. SBDs are developed.
2. SBDs are complete.
3. Software build procedures are specified.
4. SBDs include all required software units, including code and data that are part of the build.
5. Hardware and software component names and versions are described.
6. Locations of software and hardware components are described. Consideration is given for EMI/RFI and anti-virus protection of hardware and software components.
7. Methods by which the hardware components are connected together and to the sensors, actuators, and terminals are described.
8. There is documented evidence that the system made ready for verification was actually built in accordance with the SBD.
9. Software listed in the SBD is identified by version, revision, and date.
10. Software version and revision listed are the same version and revision tested.
11. The SBDs are consistent with the software specifications, as described in the SRS, software design description, and software code.
12. Consistent and uniform terminology, notation, and definitions are used throughout the SBD.
13. The SBDs specify methods to detect incorrectly built software releases.
14. The SBDs identify all errors and anomalies discovered during software build activities.

b. An integration strategy should be developed to incorporate procedures, tools and methods for managing the system/software integration effort. The strategy may be detailed in the Software Integration Plan. Verify the following, as committed to in the licensing basis:
1. Acceptance criteria is used to determine, measure and analyze the success or failure of the integration effort.
2. Integration instructions provide the technical guidance needed to carry out each integration step.
3. Procedures for the integration steps describe the input items (hardware, instrumentation, software, and data) for the steps.
4. Procedures describe the integration process for each step.
5. Procedures list the outputs of each integration step.
6. Contingency procedures/strategies are in place for an incomplete integration.

7. Procedures exist for delivering the completed integration product (System or Subsystem) to the configuration management organization.
8. Procedures exist for delivering the completed integration product (System or Subsystem) to the V&V organization for integration testing.
9. Integration tools are qualified for use.
10. Methods and controls are in place for software integration.
11. Methods and controls are in place for hardware/software integration.
12. Methods and controls are in place for system integration (if multiple vendors are involved).
13. Procedures and documentation describe all integration testing to be conducted, and expected test results. Consideration is given to Black Box/White Box testing.

A4.03.02 Integration Phase Documentation

a. Verify that development processes supporting Integration Phase activities are documented, as committed to in the licensing basis. Verify the following:

    1. Audit requirements are specified.
    2. Design Review process is defined and implemented.
    3. Configuration Management (change control) process is defined and implemented.
    4. Independent Verification and Validation (IV&V) process is defined and implemented.

b. Verify the following documentation exists for the Integration Phase activities, as committed to in the licensing basis:

    1. Integration Phase Audit Report.
    2. Integration Phase condition reports and associated corrective action.
    3. Integration Phase Configuration Management report.
    4. Integration Phase IV&V Report; verify that the report confirms 100% verification of SBDs and integration elements by the V&V organization.

A4.03.03 Integration Safety Analysis

a. An Integration Safety Analysis should be performed and documented. The analysis should determine that the complete system does not violate any safety requirements and that no additional hazards have been created by the integration activity. Verify the following, as committed to in the licensing basis:

    1. Procedures exist for hazard analysis of the integration effort.
    2. Hazard analysis concludes that the integrated system does not introduce new hazards to the developmental effort.
    3. Procedures exist for risk assessment of the integration effort.

b. The risk to an incorrect integration activity is that the system will not operate as intended, and that this will not be discovered until actual operation, possibly

during an emergency. Verifying that the integration activity has been successfully completed is part of the V&V inspection, analysis, and test activities. Verify the following, as committed to in the licensing basis:

1. Integration V&V activities demonstrate that all unit and subsystem tests required by the SVVP were successfully completed.
2. Anomalies or errors found during integration tests are resolved and documented. Final integration tests should be completed and documented.
3. Reports are consistent with the Software Integration Plan.

Appendix 5 - Inspection Guide for System/Software Life Cycle -
Validation & Test Phase

A5.01 INSPECTION OBJECTIVES

Verify that the licensee's DI&C development Validation and Test Phase process and documentation are consistent with the ITAAC design commitments and acceptance criteria.

A5.02 SAMPLE SIZE

Inspection of DI&C DAC-related ITAAC will typically rely on selection of a sample of attributes for verification. Inspection of Validation and Test Phase activities will be accomplished through verification of attributes for a representative sample of documents (typically 6 – 10 of a combination of test procedures, test plans, and test reports per inspector) stemming from testing activities. Sample size should be expanded at the inspector's discretion if inspection issues or findings are identified.

A5.03 INSPECTION REQUIREMENTS AND GUIDANCE

General Guidance.

The DI&C system validation activities are the test and evaluation of the integrated system to ensure compliance with functional, performance, and interface requirements. This determination may include analysis and evaluation of products and processes.

Testing will determine whether the requirements for the DI&C system are complete and correct. Accordingly, V&V will determine whether the "products" of each life cycle development phase implement the requirements to meet the criteria imposed by the previous life cycle phase, and that the final DI&C system complies with specified requirements. The V&V effort is governed by the V&V Plan, and activities are typically documented as follows for each life cycle phase:

- V&V Task Activity Reports
- V&V Summary Reports

Techniques and methods for V&V include document review, design review, requirements traceability, and testing. A Requirements Traceability Matrix (RTM, or equivalent tracing tool) may be developed. The RTM should depict every distinct DI&C developmental requirement and sub-requirement, and what portion of the software requirement, software design description, detailed design (code), and test will address each distinct requirement.

Inspection Requirements.

A5.03.01 Verification and Validation (V&V)

a. Metrics/indicators should be developed to determine overall effectiveness of the V&V effort. Verify the following V&V attributes, as committed to in the licensing basis:

1. Criteria are used to verify the completion of each V&V task.
2. Evaluation criteria should be provided for test plans, test specifications, test procedures and test cases.
3. Evaluation criteria should be provided for review plans, review specifications, and review procedures.
4. The error rate found during software reviews and software testing should be measured, recorded, analyzed and reported. Note - the FMEA should have identified potential errors.

b. Procedures should be developed to govern each V&V task. Verify procedures are in place for the following, as committed to in the licensing basis:

1. V&V task performance, including assumptions for each task.
2. Evaluation of design outputs.
3. Analysis and reassessment of errors detected during the V&V effort.
4. Risk assessment for DI&C developmental activities.
5. Assessment and management of changes in DI&C developmental activities.
6. V&V reporting practices.
7. V&V testing and test documentation, including testing plans, specifications, procedures and cases.

A5.03.02 System/Software Testing

a. Activities include unit testing, integration (subsystem) testing, validation testing, installation (acceptance) testing, and the regression testing of modifications. Test procedures include test documentation requirements, readiness and evaluation criteria, error reporting, and anomaly resolution requirements. Verify that the testing process and documentation includes the following attributes, as committed to in the licensing basis:

1. Documentation includes test item descriptions, test data, and test logs.
2. Test plans and procedures identify test personnel.
3. Results documentation includes types of observations, results, acceptability, and actions taken in connection with any deficiencies.
4. Software testing process includes one or more tests for each requirement in the software requirement specification (SRS), as well as the acceptance criteria for each test.
5. The result of each test clearly shows that the associated requirement has been met.
6. Test procedures contain detailed information for the test setup, input data requirements, output data expectations, and completion time.
7. Provisions in place for remediation testing, as necessary.

b. Testing for DI&C systems includes software testing, software integration testing, software qualification testing, system integration testing, and system qualification testing. The objective of Test V&V is to ensure that the software requirements and system requirements allocated to software are validated by execution of integration, system, and acceptance tests. Verify that Test V&V includes the following activities, as committed to in the licensing basis:

1. Traceability analysis using the RTM.
2. Integration V&V test procedure generation and execution.
3. System V&V test procedure generation and execution.
4. Acceptance V&V test procedure generation and execution.
5. Hazard, Risk and Security analyses.

A5.03.03 Validation & Test Phase Safety Analysis

a. Analysis should be performed of software testing to show test coverage for all software safety requirements. Analysis should also be performed on testing results for safety-critical design elements.

b. Analysis should include the following, as committed to in the licensing basis:

1. Relationship between each test and the safety requirement that the test supports.
2. Evidence to determine whether each software safety requirement has been satisfactorily tested.
3. Assessment of risk associated with implementation as indicated by test analysis.
4. Recommendation as to whether or not adequate testing has been performed.

A5.03.04 Documentation

a. Documents that summarize V&V activities, including analyses and test plan development, should be generated. Verify a sample from the following, as committed to in the licensing basis:

1. Requirements Phase V&V includes:
   - Traceability analysis
   - Software requirements evaluation
   - Interface analysis
   - Criticality analysis
   - System V&V test plan generation
   - Acceptance V&V test plan generation
   - Configuration management assessment

2. Design Phase V&V includes:
   - Traceability analysis
   - Software design evaluation
   - Interface analysis
   - Criticality analysis
   - Component V&V test plan generation
   - Integration V&V test plan generation
   - Component V&V test design generation
   - Integration V&V test design generation
   - System V&V test design generation
   - Acceptance V&V test design generation

3. Implementation Phase V&V includes:
   - Traceability analysis
   - Source code and source code documentation evaluation
   - Interface analysis
   - Criticality analysis
   - Component V&V test case generation
   - Integration V&V test case generation
   - System V&V test case generation
   - Acceptance V&V test case generation
   - Component V&V test procedure generation
   - Integration V&V test procedure generation
   - System V&V test procedure generation
   - Component V&V test execution

b. Reports that detail V&V activities should be developed. Verify the following, as committed to in the licensing basis:
   1. V&V Task Reports are developed that provide details of evaluations and analyses (activities) completed during each life cycle phase. Task Reports should also include V&V safety assessments performed as stipulated by the Software Safety Plan.
   2. V&V Summary Reports are developed for the Requirements, Design, and Implementation (Detailed Design) Phases.
   3. V&V Final Report details the validation testing that was completed, problems encountered, and disposition of issues.

Appendix 6 - Inspection Guide for System/Software Life Cycle - Installation Phase

A6.01 INSPECTION OBJECTIVES

Verify that the licensee's DI&C development Installation Phase process and documentation are consistent with the ITAAC design commitments and acceptance criteria.

A6.02 SAMPLE SIZE

Inspection of DI&C DAC-related ITAAC will typically rely on selection of a sample of attributes for verification. Inspection of Installation Phase activities will be accomplished through verification of attributes for a representative sample of installation configuration items (12 - 15 installation configuration items per inspector). Traceability (through thread audit techniques) should be verified for the same representative sample of installation configuration items. Sample size should be expanded at the inspector's discretion if inspection issues or findings are identified.

A6.03 INSPECTION REQUIREMENTS AND GUIDANCE

General Guidance.

In installation and checkout, the software product is installed and tested in the target environment. Part of the V&V activity supports the software system installation activities. The objective of installation and checkout V&V is to verify and validate the correctness of the software installation in the target environment.

Installation activities include:

- Installation Configuration Table development
- Installation Configuration audit
- Installation checkout
- Safety Analysis
- Acceptance Testing

Inspection Requirements.

A6.03.01 Installation Configuration Tables

a. Installation Configuration Tables (ICTs) should be produced. They should include functional and process characteristics to ensure that the software will be correctly configured in the operating DI&C system. Verify the following for the ICTs, as committed to in the licensing basis:

   1. ICTs configure the installed system to have the functionality that is required for the plant.

2. ICTs are consistent with the software specifications, as described in the SRS, software design description, software code, and SBDs.
3. ICTs contain all target environment (plant-specific) data.
4. ICTs are traceable; installed elements can be traced to the integrated software elements that created that installed program element.
5. ICTs allow for analysis, review, or test of each installed software system installed element.

b. Any software item which is changeable should have the intended configuration recorded in the ICT. Verify a sample of configuration item setpoints agree with values determined from setpoint calculations, as committed to in the licensing basis:

A6.03.02 Installation Activities

a. Installation configuration audit; verify the following, as committed to in the licensing basis:

1. All software products required to correctly install and operate the software are present in the installation package.
2. Installation procedures are developed and validated.
3. Installation test documentation is developed and validated.
4. Anomaly and error reporting procedures are developed.

b. Installation Checkout; verify the following are performed as part of installation, as committed to in the licensing basis:

1. Analyses or tests to verify that the installed software corresponds to the software subjected to V&V.
2. Verification that the software code and databases initialize, execute, and terminate as specified.
3. Verification that, in the transition from one version of software to the next, the software can be removed from the system without affecting the functionality of the remaining system components.

c. User documentation; verify that Operations, Maintenance and Training Manuals are developed for the DI&C system software.

A6.03.03 Safety Analysis

a. Verify that a Hazard Analysis has been performed for the installation activities.
b. Verify the performance of a security analysis, and that the security analysis results conclude that the installed software does not introduce new or increased vulnerabilities or physical and/or SDOE risks to the overall DI&C system.
c. Verify that a risk assessment has been performed to provide recommendations to eliminate, reduce, or mitigate risks associated with the installed software.

d. Verify that corrective actions have been initiated for any deficiencies identified as a result of the Hazards analysis, Security analysis, and Risk assessment.

A6.03.04 Installation Testing

a. The installation (acceptance) test activities should document the test plan, test configuration, the required inputs, expected outputs, the steps necessary to execute the test, and the acceptance criteria for each test. Verify the following, as committed to in the licensing basis:
   1. Acceptance Test procedures are developed.
   2. Issues identified during the test activity, and any action items required to mitigate or eliminate each issue, are documented. Installation problems and their resolution should be documented.

b. Test Reporting; an Acceptance Test Report should be produced describing the execution of the plan and summarizing the results. Verify the following, as committed to in the licensing basis:
   1. The Report contains a statement that the plan was successfully executed, and the system is ready for operation.
   2. The Report should document that the DI&C system operates correctly and is identical to the system that was validated during the validation phase.
   3. The report should summarize the test results after all problems have been satisfactorily resolved.

Attachment 1 - Revision History Sheet for IP 65001.22

INSPECTION OF DIGITAL INSTRUMENTATION AND CONTROL (DI&C)
SYSTEM/SOFTWARE DESIGN ACCEPTANCE CRITERIA (DAC)-RELATED ITAAC

Commitment Tracking Number	Issue Date	Description of Change	Training Needed	Training Completion Date	Comment Resolution Accession Number
N/A	12/19/11 CN 11-041 ML112560050	New	None.	N/A	ML112550611