1. Acceptance Criteria for Design Implementation

The acceptance criteria address specific software or logic life cycle process implementation activities and documentation. These activities and products, when found to be acceptable, provide the reviewer with confidence that the plans have been carried out.

The reviewer confirms that the plans have been followed by the software or logic developer. The detailed acceptance criteria are provided by the developer and evaluated by the NRC staff in its acceptance of the plans. In addition to verifying that plans have been followed, the reviewer should pay particular attention to the following areas.

- Safety Analysis Activities
- Verification and Validation Activities
- Configuration Management Activities
- Testing Activities

1.1. Acceptance Criteria for Safety Analysis Activities

The safety plan describes the safety analysis implementation tasks that are to be performed. The acceptance criterion for safety analysis implementation is that the tasks in that plan have been carried out in their entirety. Documentation exists to show that the safety analysis activities have been successfully accomplished for each life cycle activity group and that the proposed digital system software or logic is, in fact, safe. In particular, the documentation should show that: the system safety requirements have been adequately addressed for each activity group; that no new hazards have been introduced; that the software or logic requirements, design elements, and code elements that can affect safety have been identified; and that all other software or logic requirements, design, and code elements will not adversely affect safety. The safety analysis activities should also include assessment of the SDOE risk, including an assessment of the measures used to ensure that the design products do not contain undocumented code, and other unwanted or undocumented functions or applications.

1.2. Acceptance Criteria for Software or Logic Verification and Validation Activities

The verification and validation (V&V) plan describes the V&V implementation tasks that are to be carried out. The acceptance criterion for software or logic V&V implementation is that the tasks in the V&V plan have been carried out in their entirety. Documentation should exist that shows that the V&V tasks have been successfully accomplished for each life cycle activity group. In particular, the documentation should show that the requirements, design, code, integration, and installation design outputs satisfy all specified system requirements.

Problems identified by the V&V effort should be documented, together with any action items required to mitigate or eliminate each problem. A record should be kept of actions taken in response to the action items and the appropriate configuration management activities should be performed.

<u>Requirements Traceability:</u>

As part of the software or logic V&V effort, traceability analyses should be performed and documented at specified intervals. This traceability analysis documentation should clearly show the linkage between each requirement imposed on the software or logic by the system requirements document and system design documents. The analysis documentation should allow traceability in both directions. It should be organized so that as design, implementation,

and validation take place, traceability information can be added for these activities. It should be updated at the completion of each life cycle activity group. The final analysis documentation should permit tracing from the system requirements and design through the software or logic requirements, design, implementation, integration, validation, and installation. The integration V&V activities should demonstrate that all unit and subsystem tests required by the V&V plan were successfully completed. The final integration V&V report should describe the procedures followed and the tests performed during integration. This report should be consistent with the integration planning document.

Testing:

The V&V activities should demonstrate that all validation and acceptance tests required by the V&V plan were successfully completed. All test activities should document the test configuration, the required inputs, expected outputs, the steps necessary to execute the test, and the acceptance criteria for each test. The testing process should contain one or more test cases for each requirement in the applicable requirements specification and the result of each test should clearly show that the associated requirements have been met.

Procedures for handling errors and anomalies encountered during the testing should also be reviewed. These procedures should include correction procedures (including configuration management), and provision for re-test to ensure the problems are resolved. A final report summarizing the V&V testing should be provided. The report should contain a statement that the testing was successful and that the software or logic tested met all of the requirements of the requirements specification.

Thread Audit

One of the accepted methods of checking the V&V effort is to perform a “thread audit.” This consists of picking a sample of plant parameters and tracing the software or logic implementation of these parameters from the purchase specification and development of the functional requirements to the writing and testing of the code. The sample size should be sufficiently large to ensure a representative sample of the requirements and of the software or logic code have been reviewed. The minimum sample size should be determined by statistical significance criteria. This review includes:

1) Reviewing actual sections of the code on a sample basis. Since the reviewer is seldom an expert in a particular language, this may necessitate that the responsible design engineer walk the reviewer through the code. If the reviewer is unable to follow this explanation, this portion of the thread audit should be delegated to more experienced staff personnel or an independent contractor.
2) Examining the various levels of software or logic development documents and comparing them to the code.
3) Examining problem reports and test plans for the selected requirements, and verifying the corrections made. It would be unusual if there were no problem reports, and if this is the case, the testing and review procedures should be carefully examined to ensure that a thorough test and review was performed.
4) Each of the completed problem reports should show what was done to resolve the problem, and how that resolution was tested. This would also be a good time to check the configuration management procedures to see how the revised code was put under configuration management.

5) Examining the engineering cross-discipline interfaces to ensure that nuclear specific needs were correctly incorporated into the design.
6) Examining the applicant or licensee interface to ensure plant specific requirements are correctly incorporated.
7) Ensuring that the V&V process is followed according to the vendor's plan.
8) Reviewing the final results of the process.

Items other than requirements may be included in the thread audit, depending on the software tools used and the exact nature of the programing requirements and the methods. The reviewer should not hesitate to ask why something was done in a particular manner, and should use experience and judgment to assess the results.

If errors are found by the review staff, the appropriate V&V records should be examined to see if the V&V team has also caught the errors. If the requirements, code and test have been verified and validated without finding the error, this may be indicative of a quality problem. Additional requirements should then be checked to see if this is a systematic or an isolated problem. If several of these problems are found, the adequacy of the software or logic development process may be insufficient to produce high quality software or logic for use in safety-related applications in nuclear power plants. The reviewer should discuss these concerns with management to determine appropriate corrective actions including process improvements.

A successful thread audit necessitates that the reviewer be familiar with the various plans and procedures used for the software or logic development and life cycle, and the various NRC requirements, NUREGs, RGs, and industry standards that those plans and procedures are based upon. This knowledge is necessary to be able to determine if the methodologies examined during the thread audit are those which the developer committed to using, and if they are being used correctly. The thread audit presents an opportunity for the reviewer to examine the actual development process rather than the documented plans for a development process.

1.3. Acceptance Criteria for Software or logic Configuration Management Activities

Software or logic configuration management can be partitioned into two types of activities: the management and control of the software or logic and the associated development environment; document control. There are many configuration management tools available for software and logic development. A particular tool should be selected, evaluated, and used properly for software configuration control.

The development plan describes the software or logic and documents that will be created and placed under configuration control. If the software or logic and documents are controlled in different systems, then each should be described, as well as their relationship to each other. The configuration management plan describes the implementation tasks that are to be carried out. The acceptance criterion for software or logic configuration management implementation is that the tasks in the configuration management plan have been carried out in their entirety.

Documentation should exist that shows that the configuration management tasks for that activity group have been successfully accomplished. In particular, the documentation should show that configuration items have been appropriately identified; that configuration baselines have been established for the activity group; that an adequate change control process has been used for changes to the product baseline; and that appropriate configuration audits have been held for the configuration items created or modified for the activity group.

Each configuration item should be labeled unambiguously so that a basis can be established for the control and reference of the configuration items defined in the SCMP. Configuration

baselines should be established for each life cycle activity group, to define the basis for further development, allow control of configuration items, and permit traceability between configuration items. The baseline should be established before the set of activities can be considered complete. Once a baseline is established, it should be protected from change. Change control activities should be followed whenever a derivative baseline is developed from an established baseline. A baseline should be traceable to the baseline from which it was established, and to the design outputs it identified or to the activity with which it is associated.

Configuration control actions should be used to control and document changes to configuration baselines. A configuration control board (CCB) should exist with the authority to authorize all changes to baselines. Problem reports should be prepared to describe anomalous and inconsistent software or logic and documentation. Problem reports that require corrective action should invoke the change control activity. Change control should preserve the integrity of configuration items and baselines by providing protection against their change. Any change to a configuration item should cause a change to its configuration identification. This can be done via a version number or attached change date. Changes to baselines and to configuration items under change control should be recorded approved and tracked. If the change is due to a problem report, traceability should exist between the problem report and the change. Software or logic changes should be traced to their point of origin, and the software or logic processes affected by the change should be repeated from the point of change to the point of discovery. Proposed changes should be reviewed by the CCB for their impact on system safety.

Status accounting should take place for each set of life cycle activities at the completion of those activities. The status accounting should document configuration item identifications, baselines, problem report status, change history and release status. The configuration management organization should audit life cycle activities to confirm that configuration management procedures were carried out in the life cycle process implementation.

1.4. Acceptance Criteria for Testing Activities

Thorough software or logic testing consists of testing the smallest testable units, and then integrating those units into larger testable units, and testing that integrated unit. This process is repeated until the system is tested after installation.

2. Acceptance Criteria for Design Outputs

Design outputs are the products of a design process. Design outputs can include software, logic, configuration files, intermediate product support files such as binary files or compilation reports, and design documentation such as drawings, diagrams, and reports. Each I&C safety system will have a unique set of design outputs that are defined by the platform being used, the processes for performing design and the application specific design requirements. The following is a list of example design outputs that could be the products of a design effort. A suitable list of design outputs should be developed for each safety system application.

Software or logic Requirements Specification
Software or logic Architecture Description
Software or logic Design Specification
Code Listings
System Build Documents
Installation Configuration Tables
Operations Manuals
Software or logic Maintenance Manuals
Software or logic Training Manuals

2.1. Requirements Activities - Software or logic Requirements Specification

Errors in requirements or misunderstanding of requirement intent are a major source of software or logic errors. The requirements should be carefully examined by the reviewer. If the requirements are not clear to the reviewer, they will probably not be clear to the software or logic design team.

The thread audit is a tool which can be used to check effectiveness of requirements traceability. During the thread audit, for each requirement traced, the reviewer should check that each requirement is complete, that the requirements are consistent with the overall safety system requirements, and that the requirement is not in conflict with some other requirement.

The requirements should be understandable and unambiguous. Each requirement should be traceable to one or more safety system requirements, and the requirements traceability matrix should show where in the software or application logic the required action is being performed. The requirements traceability matrix should also show where the particular requirement is being tested.

2.2. Design Activities - Software or logic Architecture Description

The reviewer should be able to refer to this architecture to understand how the software or logic works, the flow of data, and the deterministic nature of the software or logic. The architecture should be sufficiently detailed to allow the reviewer to understand the operation of the software or logic.

2.3. Design Activities - Software or logic Design Specification

The SDS is primarily used to ensure that the software or logic code accurately reflects the software or logic requirements. The thread audit should check several of the requirements and follow them through the final code, but the entire SDS should be read by the reviewer to determine that it is understandable, and contains sufficient information. In addition, the V&V report on the software or logic design specifications should be carefully reviewed. The reviewer

should not be able to find any problems which have not been found and documented by the V&V team. If the software or logic design specifications is reviewed after completion of the V&V effort, the reviewer should find no errors.

2.4. Implementation Activities - Code Listings

The code listings should have sufficient comments and annotations that the intent of the code developer is clear. This is not only so the reviewer can understand and follow the code, but also so future modifications of the code are facilitated. Undocumented code should not be accepted as suitable for use in safety-related systems in nuclear power plants. The documentation should be sufficient for a qualified software or logic engineer to understand. If the reviewer does not have enough experience in this particular language or with the software tool being used, the reviewer may require the assistance of other NRC personnel or independent contractor personnel to make this determination.

2.5. Integration Activities - System Build Documents

The system build documents are needed to verify that the software or logic actually delivered and installed on the safety system is the software or logic which underwent the V&V process and was tested. Any future software or logic maintenance will depend on the maintainers knowing which version of the software or logic to modify. The reviewer should check to ensure that the software or logic listed in the build documentation is identified by version, revision, and date. The reviewer should also verify that this is the version and revision which was tested. This information should all be available from the configuration management group.

2.6. Installation Activities - Installation Configuration Tables

In the event that the software or logic has options for use, variable setpoints or other data, or may operate in various methods, the software or logic needs to be configured for the particular plant requirements. Any software or logic item which is changeable should have the intended configuration recorded in the installation configuration tables, and the reviewer should sample these configuration items to verify that they are correct. The reviewer should verify that the V&V team has already made this determination, and should then sample various items. For example, the reviewer could verify that setpoints shown in the installation configuration tables agree with values determined by setpoint calculations. Other configuration items may require reference back to the original system specification. The installation configuration tables are a critical item for configuration management, both by the vendor and by the applicant or licensee.

2.7. Installation Activities - Operations Manuals

The reviewer should keep in mind that the intent of the staff review of manuals is to ensure that digital system is safe, and therefore the portion of the operations manuals which is of primary concern is the portion which deals with operation of the system under unusual or emergency conditions. The portion of the manual which deals with normal operation should be reviewed, but does not require the depth of review that emergency operations does.

2.8. Installation Activities - Software or logic Maintenance Manuals

Prior to the review of the maintenance manuals, the reviewer should determine if maintenance will be done by applicant or licensee or vendor personnel. In many instances, the applicant or licensee only performs maintenance to replace failed circuit boards, and the boards are then sent to the vendor for repair. In this instance, the maintenance manuals used by applicant or licensee personnel do not need to be as detailed as would be required if the applicant or licensee was doing board level repairs. The reviewer should use judgment to determine how adequate the level of detail is in each of the maintenance manuals.

2.9. Installation Activities - Software or logic Training Manuals

The reviewer should determine that the training manuals are both understandable and useful. One method which can be used for this determination is for the reviewer to take the training courses which use these manuals. The training manuals will generally be aimed at either the technician level or the software or logic engineer, and therefore the determination of the understandability and usability of the training manual needs to take into account the intended use. If the reviewer does not have enough experience with training or training documentation to make this determination, the reviewer may require the assistance of other NRC personnel or independent contractor personnel.