Docket

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D. C. 20555

December 11, 1992

Docket Nos. 50-259, 50-260,
and 50-296

Tennessee Valley Authority
ATTN: Dr. Mark O. Medford, Vice President
Nuclear Assurance, Licensing and Fuels
3B Lookout Place
1101 Market Street
Chattanooga, Tennessee 37402-2801

Dear Dr. Medford:

SUBJECT: DRAFT GUIDELINES FOR AUDITING DIGITAL SAFETY SYSTEMS - BROWNS FERRY UNITS 1, 2, AND 3 (TAC NOS. M84161, M84162, AND M84163)

Enclosed for your information are draft guidelines developed by the NRC's contractors for conducting audits of safety system software verification and validation (V & V) programs. These guidelines are in draft form and have not as yet been endorsed by the NRC staff.

During a conference call with Tennessee Valley Authority (TVA) and General Electric (GE) on November 19, 1992, the staff solicited TVA's interest in participating in a pilot project that would exercise the enclosed guidelines. The NRC staff and its contractors would like to examine these guidelines during an actual audit of the Nuclear Measurement Analysis and Control (NUMAC) Reactor Building Ventilation Radiation Monitoring (RBVRM) system being installed at the Browns Ferry Nuclear Plant (BFN). An audit plan would be prepared and implemented by the staff and its contractors using recommendations from these guidelines as part of the staff's evaluation of TVA's license amendment application of July 23, 1992. You are requested to notify the staff (verbally is acceptable) by December 18, 1992, if TVA intends to participate as a pilot plant. If TVA does not choose to participate, the staff will perform its audit of the NUMAC RBVRM at BFN using previously developed guidance described in prior safety evaluations of the Eagle-21 system issued for Sequoyah and Zion nuclear plants.

DF01

In either case, the staff intends on conducting an audit of the NUMAC RBVRM at GE's facilities in San Jose, California, during the week of January 11, 1993, and at BFN during the week of January 18, 1993. If TVA or GE cannot support the staff's audit activities during these weeks, please notify the NRC as soon as possible to coordinate new dates. Any questions regarding this letter or the enclosed guidelines should be directed to Thierry M. Ross, NRC Project Manager at (301) 504-1474.

Sincerely,

Original signed by

Frederick J. Hebdon, Director
Project Directorate II-4
Division of Reactor Projects - I/II
Office of Nuclear Reactor Regulation

Enclosure:
As stated

cc w/enclosure:
See next page
Distribution
Docket File
NRC & Local PDR
BFN Reading
S. Varga 14-E-4
G. Lainas 14-H-1
TRoss/JWilliams/CBeardslee
F. Hebdon
M. Sanders
OGC 15-B-18
ACRS(10)
B. Wilson RII
E. Merschoff RII

OFC	PDII-4/LA	PDII-4/PM	PDII-4/PM	PDII-4/D	
NAME	MSanders	TRoss:as	JWilliams	FHebdon	
DATE	12/11/92	12/11/92	12/11/92	12/11/92	

Tennessee Valley Authority
ATTN: Dr. Mark O. Medford

Browns Ferry Nuclear Plant

cc:

Mr. John B. Waters, Chairman
Tennessee Valley Authority
ET 12A
400 West Summit Hill Drive
Knoxville, Tennessee 37902

Mr. J. R. Bynum, Vice President
Nuclear Operations
3B Lookout Place
1101 Market Street
Chattanooga, Tennessee 37402-2801

Site Licensing Manager
Browns Ferry Nuclear Plant
Tennessee Valley Authority
P.O. Box 2000
Decatur, Alabama 35602

Mr. O. J. Zeringue, Vice President
Browns Ferry Nuclear Plant
Tennessee Valley Authority
P.O. Box 2000
Decatur, Alabama 35602

Mr. M. J. Burzynski, Manager
Nuclear Licensing and Regulatory Affairs
5B Lookout Place
Chattanooga, Tennessee 37402-2801

TVA Representative
Tennessee Valley Authority
11921 Rockville Pike
Suite 402
Rockville, Maryland 20852

General Counsel
Tennessee Valley Authority
ET 11H
400 West Summit Hill Drive
Knoxville, Tennessee 37902

Chairman, Limestone County Commission
P.O. Box 188
Athens, Alabama 35611

State Health Officer
Alabama Dept. of Public Health
434 Monroe Street
Montgomery, Alabama 36130-1701

Regional Administrator
U.S.N.R.C. Region II
101 Marietta Street, N.W.
Suite 2900
Atlanta, Georgia 30323

Mr. Charles Patterson
Senior Resident Inspector
Browns Ferry Nuclear Plant
U.S.N.R.C.
Route 12, Box 637
Athens, Alabama 35611

Site Quality Manager
Browns Ferry Nuclear Plant
Tennessee Valley Authority
P. O. Box 2000
Decatur, Alabama 35602

4.3 PREAUDIT ACTIVITIES

This section provides general guidance for preparing for audits of digital safety systems and is based primarily on techniques used by NRC in the past. An audit (or series of audits) should be performed to determine the applicant's conformance to current NRC general design criteria, requirements, and applicable regulatory guides as identified in *Standard Review Plan for the Review of Safety Analysis Reports for Nuclear Power Plants* (NRC 1981). In addition, the review team should be familiar with accepted related industry standards. Although these standards may not have been adopted as official NRC standards, deviations from their provisions should be noted in the audit report. Each auditor should also be prepared to comment on perceived discrepancies and deficiencies in areas of his expertise that are not specifically identified by standards and regulatory guides. The reviewer should expect to find documentation relating to each audit heading appearing in Tables 4.1 through 4.5 in subsequent sections of this report. Because acceptance criteria are based on the applicant's meeting the relevant requirements, the elements of the audit plan in these tables are presented in a format intended as a guide to the requirements that ANSI/IEEE-ANS-7-4.3.2-1982 places on the applicant/vendor.

All audit procedures recommended below closely follow IEEE Std. 1028 (1988), Reviews and Audits. Multiple audits may be required to cover all elements of the V&V program. The objective of preaudit activities is to ensure that both the applicant/vendor and the audit team are well prepared for the audit. The recommended preaudit steps are listed below.

1. Appoint an audit team with backgrounds that ensure an adequate review, and identify the team leader.

2. Prepare an audit plan using Tables 4.1–4.5 (presented later) and other checklists in this report as a guide and an agenda stating time and place, and forward it to the applicant. The scope of the audit (which depends on the life cycle phase of the audit) should be clearly defined and should include (a) documents to be provided at audit; (b) a list of applicant personnel required to support the audit (e.g., designers, programmers, verifiers, validators); (c) if appropriate to this audit, the identity of one string (or thread path) to be audited (see discussion in last paragraph of this section).

3. Request submittal of the V&V plan prior to the first audit, with enough time allowed for the audit team to generate questions and comments.

4. Review the V&V plan (on site if not previously provided) for (a) proposed V&V activities [refer to ANSI/IEEE Std. 1012 (1986), "Software Verification and Validation Plans" for guidance, and to Sect. 4.5 below]; (b) independence of V&V personnel from design personnel; (c) procedures and practices of the V&V team; and

(d) documentation to be provided (e.g., trouble reports, test reports, failure analysis reports).

In the past NRC has used the string (or thread path) audit for safety systems. The procedure is to select a sensor signal and follow the signal from that sensor through all hardware and software components up to the system interface with the final output devices. In evaluating the processing of the selected signal, the audit team should address all elements of the software model, conformance to ANSI/IEEE-ANS-7-4.3.2-1982, and all applicable elements of the V&V plan. It is most useful when (1) the audit team participates in selection of the string and (2) a string is selected that involves software for which a trouble report and subsequent resolution report have been filed. If properly applied, the string audit gives insight into system documentation, traceability, the requirements matrix, the V&V process, and the failure reporting system. Choosing a string in which a problem has been encountered allows examination of procedures for reverifying and revalidating the software.

4.4 AUDIT ACTIVITIES

This section provides guidance on setting the agenda for an audit. The agenda described below has been used successfully in the past by NRC. It may be modified as appropriate.

1. The audit team leader holds an entry briefing to introduce all audit team members and vendor and applicant personnel, to state the scope of the audit with references to the audit plan submitted earlier, and to agree on the audit schedule.

2. The audit team then executes the audit plan through briefings by the applicant/vendor, by examination of all documentation specified in the audit plan, and by performing the string check (if required by this audit) using audit guidelines (see discussion in Sect. 4.2.3) and engineering judgment. During the string check and throughout the audit, the audit team should note all discrepancies and areas of concern.

3. At the conclusion of the audit examinations, the audit team holds a caucus to prepare an exit briefing. The team leader compiles all audit team concerns and any noted deviations from the standards. The audit team must then reach agreement on acceptable elements of the audit, unacceptable elements of the audit, severity of the discrepancies noted, and overall concerns (see Sect. 4.4.1 for guidance).

4. The team leader holds an exit briefing with the audit team and applicant/vendor personnel in attendance. During this briefing, the team leader reiterates the scope of the audit and states that the audit plan and audit results will serve as the basis for the audit report, with the proviso that the preliminary results are subject to NRC management review and approval. The team leader then states the preliminary results of the audit including acceptable elements, unacceptable elements in terms of

nonconformance to regulations (generally ANSI/IEEE-ANS-7-4.3.2-1982), discrepancies noted, and areas of concern. The team leader provides an estimate of the date of safety evaluation report (SER) publication and discusses the next audit (if required) including proposed schedule and scope.

4.4.1 Guidance for Acceptance or Rejection of Reviewed Software

Because the judgment of the quality of each element of a V&V program is likely to be at least partially subjective, it is inappropriate to average the grades and indicate an overall score that indicates pass or fail. Rather, it is appropriate to consider each major element prejudged to be essential to approval and ensure that the audit determined all elements to be satisfactory. If not, the vendor/designer/utility should be instructed to correct the problems and submit to further review. When less essential elements of the design or program deviate from "standard practice" but there was still adequate assurance that the final product was satisfactory, these elements should be flagged and considered (negatively) in the overall evaluation. In such cases the audit team must use its knowledge and experience to judge the severity of the deviations.

4.5 VERIFICATION AND VALIDATION PLANNING

Experience with large software development programs has shown that V&V planning should start early and use skilled and experienced teams. Good communications between individual team members and between developers and users, along with careful documentation, should be maintained throughout the process. Maintaining communication and documentation is especially important in the transitions between stages of the system's life cycle. Frequent informal reviews should be used to supplement infrequent formal ones. Special efforts should be made to issue clear, high-quality reports on the tracking of errors and subsequent corrective actions.

Two other software V&V recommendations from ANSI/ASME Std. NQA-1, part 2.7 (1987) are applicable: First, the standard states that the V&V activities must be planned and performed for each system configuration that may affect the software and that review activities must be performed by individuals other than those who designed the system. [Other potential second-party reviewers could be the user (utility) or a software company specializing in V&V.] Second, communications between the design and review groups should be documented in written reports.

The requirements for V&V testing and reporting should be stated with appropriate documentation. Properly performed V&V increases the confidence level that a system will operate according to the functional requirements and is essential to the qualification process. Development of a validation test plan is an essential step in the V&V process, and this plan should be designed to demonstrate that the completed system meets all system requirements. Tests requiring selected combinations of steady state inputs (static tests), time-varying inputs (dynamic tests), functional tests, and statistical tests can be used to challenge various features and capabilities of the design. Error injection tests (to test

the tests) also may be used. Ideally, the tests should use preprogrammed data input tapes and/or analog and digital simulations of the signals used in the actual system.

The test plan should specify the criteria used to determine whether the system passes or fails and should determine also a "grade" to indicate how close the system comes to perfection or failure. The details of the test specifications including, for example, the test environment and special equipment should be carefully documented and recorded along with the results (including raw data), analyses, and nonconformance reports. A test plan includes both the (factory) acceptance tests that grade the system on how well it meets the design requirements and the field tests that verify proper installation of the validated system.

An auditor's checklist for a review of the V&V plans and the evaluation of V&V activities (indicated in Tables 4.1–4.5) could be developed from the following list of questions.

1. What standards and guidelines were used in the V&V and QA planning? Do the project management and the design team members demonstrate an adequate familiarity with the standards?

2. Does the documentation available on the project indicate that V&V planning began at an early stage and was maintained throughout?

3. Does the documentation show that good communications (e.g., via formal or informal reviews, internal reports, and meetings) began early and were maintained?

4. Has an appropriate means of configuration control and a labeling system been established and followed such that software changes are documented and validated properly?

5. Do the records show that a reasonable effort was made to design for functional diversity? What independent groups of experts were used for verification?

6. Does the V&V team have organizational independence from those responsible for system design? Does the V&V team have technical qualifications comparable to those of the design team?

7. What means have been used in the test plans to demonstrate conformance with the requirements?

8. Has a satisfactory combination of test methods (e.g., static, dynamic, statistical, error injection, structural, functional) been specified?

9. Are the metrics used to determine the bottom-line results of the tests (pass/fail/ragged edge) satisfactory and mutually agreeable (preferably in advance) by both the designer and V&V personnel?

10. Are sufficient check procedures in place to ensure that systems in which errors are detected are properly analyzed, reported, corrected, and (if necessary) retested?

11. Are technical reviews and audits themselves well documented? ANSI/IEEE-ANS Std. 7-4.3.2-1982 lists the (minimum) documentation required, in the form of a written report: (a) objective; (b) criteria to meet the objectives; (c) personnel qualifications; (d) preverification activities; (e) agenda, schedule, and list of available data and documentation; and (f) description of the design activities that are affected by the verification effort.

12. Have all required V&V activities been scheduled?

4.6 SAFETY SYSTEM REQUIREMENTS

The specification and review of the system requirements have proved to be crucial elements of most major software-intensive projects. The requirements must be shown to embody the necessary and sufficient conditions for safety system functions and, at the same time, must result in practical solutions to the right problems. Table 4.1 shows the portion of the elements of the audit plan to which these concerns and checklist items apply.

Because the system requirements have the widest audience, they should be crafted carefully to ensure that design team members, advisors, and reviewers of widely varying disciplines are able to understand them. Hence, they should have the attributes of clarity, consistency, traceability, and supportability. They should also be presented in a hierarchical form, representing the logic, timing considerations, and mathematical relations involved.

The bases for each requirement should be clearly documented. Records are especially crucial for later in the life cycle, when the original team may be largely disbanded and the need for changes in the system arises. The life cycle (from initial design to decommissioning) of a reactor subsystem such as a safety system could be 50 years or longer. The next generation should know why things were done the way they were in the early stages, and thus be able to justify why changes should be made on the basis of new data or new technologies.

An important feature of the system is the testability of its requirements. Whether each requirement should be testable and to what extent should be part of the system requirement specification.

Development of an auditor's checklist for a review of the system requirements could be based on the following list of questions.

1. Do the requirements address the correct problems and have the potential for practical solutions?

2. Have the bases for each requirement been described and documented?

Table 4.1 Requirements portion of the elements of audit plan

Computer system requirements
Project description defined by utility
U.S. Nuclear Regulatory Commission regulations[a]
Hardware requirements that impact software (specifying as a minimum)[b(1)]
Input/output
Control of program and data changes
Initialization
Diagnostics
Human factors
Fault-tolerant techniques
Timing and memory margin
Interrupts
Verification of hardware requirements[b(2)]
Written documentation of discrepancies and resolutions
Software requirements (specifying as a minimum)[b(3)]
Inputs
Support software
Algorithms
Data files
Outputs
Initialization
Responses to system failures
Operator interface
Diagnostics
Timing
Idle time and excess memory
Software security
Verification of software requirements[b(2)]
Written documentation of discrepancies and resolutions
Hardware-software integration requirements[b(4)]
Integration plan
Test procedures and acceptance criteria
System test configuration
Quality assurance for integration and change control
Verification of hardware-software integration requirements[b(2)]
Written documentation of discrepancies and resolutions

[a]See, for example, "Criteria for Programmable Digital Computer System Software in Safety-Related Systems of Nuclear Power Plants," Regulatory Guide 1.152 (Task IC 127-5), U.S. Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, 1985.

[b]*Source*: Based on data from "Application Criteria for Programmable Digital Computer Systems in Safety Systems of Nuclear Generating Stations," ANSI/IEEE-ANS-7-4.3.2-1982, American Nuclear Society, La Grange Park, Illinois, 1982: (*1*) Sect. 3.1, (*2*) Sect. 7, (*3*) Sect. 3.2, (*4*) Sect. 3.3.

3. Are the requirements clearly written, unambiguous, and specific (not generalized), and have they been communicated satisfactorily to all members of the design and review teams?

4. Have the input/output and interface requirements with other equipment and systems been addressed correctly and completely?

5. Have the requirements for interfaces with operators and maintenance personnel been adequately defined along with appropriate training needs?

6. Have the V&V and QA plans been identified, defined, and implemented properly?

7. Have the requirements for operating environment and environmental qualification for external impacts (including physical such as quake, fire, moisture, or electromagnetic interference and security considerations such as computer viruses and sabotage) been defined adequately?

8. Are the requirements for testing satisfactory (i.e., would the required acceptance tests adequately demonstrate that the system will perform all required functions and not perform unintended ones even in the face of a prescribed set of failure conditions)? The capabilities matrix, when properly developed, contains information useful for evaluation. For example, the acceptance tests for each requirement should be clearly stated and should include the expected results. At a minimum, testing should be performed with input values in the expected range as well as values both above and below the expected range [see specific guidance in ANSI/IEEE Std. 829 (1983); ANSI/IEEE Std. 1008 (1987); IEC (1986); NASC-39 (1981); Saglietti (1989); Wallace (1989)].

9. Are the acceptance test criteria clear, adequate, and quantitative?

10. Do acceptance tests incorporate the use of plant or appropriate subsystem simulations?

11. Are the required provisions for on-line in-service testing and diagnostics adequate and suitably monitorable?

12. Is the configuration control adequate [i.e., are the requirements for following the required documentation (from origin through changes), error tracking/correcting, maintenance, and system upgrade processes adequate, and do they provide for documentation in a readily usable form (such as in a relational database)]?

4.6.1 Software Requirements

Section 3.2 of ANSI/IEEE-ANS-7-4.3.2-1982 requires that all software used in safety systems be documented and verified. This section includes a brief description of the documentation that should be available to the reviewer from the requirement specification process.

The software requirements may be included in an overall computer system requirements document or in a separate document.

In the discussion that follows, "Guidelines for Documentation of Computer Programs and Automated Data Systems" (NIST) has been used as the guideline for the documentation, with additional guidance from ANSI/IEEE-ANS-7-4.3.2-1982.

A software requirements description document (which may be a portion of a larger document) defines requirements (e.g. functional, fault tolerance, and performance requirements). The software requirements document, in combination with the hardware requirements document and the integration requirements document, is used as a basis for the implementation phase, development of the requirements matrix, integration planning, and validation testing. Figure 4.2 shows sample contents of a system requirements document reflecting Sects. 3.2.1 through 3.1.12 of ANSI/IEEE-ANS-7-4.3.2-1982.

A requirements matrix, indicating which system components are used to implement each requirement, is crucial to the auditing phase of qualifying a computer system in that it provides a paper trail from requirements to testing. A format often used (1) lists each requirement, (2) identifies which system component implements each requirement, and (3) points to the integration/validation test used to confirm that the requirement has been met. Figure 4.3 illustrates a sample requirements (or capabilities) matrix format. The matrix should be reviewed at all audits to ensure that the contents adequately reflect the current stage of the life cycle.

After preparation of the software requirements document, an independent verification of the requirements against the functional requirements must be made. The reviewer should examine the V&V plan to determine the procedures and personnel proposed for this phase and the evidence that this review took place. In addition, the reviewer may apply the following criteria for judging the adequacy of the software requirements specification process.

Section 4.1 of ANSI/IEEE-ANS-7-4.3.2-1982 requires a software development plan that should include the following information (at a minimum): (1) identification of the QA program used; (2) documentation of software development standards ensuring consistency, accuracy, error tolerance, and modularity; and (3) provisions for auditability and testability (e.g., documents, traceability matrix, V&V plan). In many instances, software is developed in accordance with the licensee's standard procedure manuals. In this case, all applicable documents should be available for review.

Failure reporting and analysis requirements should be provided. Particular attention should be paid to both the reporting forms and the completeness of their use as well as to the plans for analyzing faults or failures that may occur in the hardware or software. The number and types of failures and the presence or absence of trends and generic problems affect the confidence level of the safety of the system.

1. General Information
   - 1.1. Summary
   - 1.2. Environment
   - 1.3. References

2. Overview
   - 2.1. Background of functional requirements (purpose and scope)
   - 2.2. Performance objectives
   - 2.3. Existing system
   - 2.4. Proposed system
   - 2.5. Summary of impacts
   - 2.6. Cost considerations

3. Requirements
   - 3.1. Functions
     - 3.1.1. Initialization
     - 3.1.2. Application processing
     - 3.1.3. Interprocessor communication
     - 3.1.4. Exception handling
     - 3.1.5. Interfaces
   - 3.2. Performance
     - 3.2.1. Accuracy
     - 3.2.2. Timing
     - 3.2.3. Extendibility
   - 3.3. Inputs and outputs
     - 3.3.1. Inputs and outputs
     - 3.3.2. Application inputs and outputs
   - 3.4. Data characteristics
     - 3.4.1. Data (e.g., interprocessor messages, logs)
     - 3.4.2. Application data
   - 3.5. Fault Detection and recovery and fallback procedures
     - 3.5.1. Faults (single, multiple, transient, intermittent)
     - 3.5.2. Faults in application software
     - 3.5.3. Faults in controlled system

4. Operating Environment
   - 4.1. Operating system
   - 4.2. Runtime libraries
   - 4.3. Debugging and testing software
   - 4.4. Communication interfaces

Fig. 4.2 Requirements document contents.

Requirement\ Capability	(1) Requirements	(2) Design	(3) Design Verification	(4) System Test Plan	(5) Validation Results	(6) Validation Results
1. Availability of 0.99	p. 15	p. 121-130	p.14	NT	NT	NT
2. Update every 5 s	p. 18	p. 140-146	p. 18	p. 18	p. 6	p. 9
3. Human factors engineering	p. 19	p. 202-260	p. 7	NT	NT	NT

Fig. 4.3 Example of a capabilities matrix for tracking the implementation of requirements. [(1) through (6) are reference documents; NT = not to be tested.] *Source: Verification and Validation for Safety Parameter Display Systems*, NSAC-39, National Safety Analysis Center, Electric Power Research Institute, Golden, Colo., 1981.

4.7 SOFTWARE AND SOFTWARE-HARDWARE INTERFACE DESIGN

The objective of the design review is to verify that the hardware and software design will result in a safety system that unambiguously satisfies the system requirements. Software design requirements follow directly from each applicable system requirement. Specific attributes of the software pertain to its functionality, performance, design constraints, and external interfaces. Other attributes should include, for example, portability, acceptance criteria, access control, maintainability, and traceability. Table 4.2 lists the elements of the audit plan to which these guidelines apply.

During the design review, each system performance and interface requirement should be correlated with one or more specific software design features that contribute to satisfying that particular requirement. Correlation is most easily carried out on systems that use a top-down design approach in which the hierarchy of requirements is spelled out.

The use of software tools in the V&V process should be encouraged. ANSI standards do not require verification of such tools, but QA measures are recommended at a level appropriate for their importance in the overall process.

Although detailed requirements for the safety system logic and means for achieving redundancy are (appropriately) not specified in the standards, the design review should ensure that the design chosen will meet the required overall system availability and reliability goals.

Table 4.2 Design portion of elements of the audit plan review

Software development*(1)
Development plan*(2)
Organization and procedures for each phase
Development standards and procedures
Assurance of auditability and testability in all phases
Quality assurance program
Detailed design*(3)
Consistency with development methodology
Correlation with each software requirement
Verify detailed design*(4)
Design implementation*(5)
programming techniques
documentation standards
coding conventions
test requirements
Verify implementation*(4)
Test case selection criteria
Structural verification testing
Functional verification testing
Level of verification testing
Acceptance criteria
Discrepancy record format, resolution format, test report format
Reviewer should examine design and coding error summaries (i.e., review one error in depth)
Hardware-software integration*(6)
Configuration of hardware and software used in testing
Test equipment and calibration
Simulation models used
Test results
Discrepancy reporting and corrective actions
Verification of hardware-software integration*(4)
Written documentation of all discrepancies and resolutions

*Source: Based on data from "Application Criteria for Programmable Digital Computer Systems in Safety Systems of Nuclear Generating Stations," ANSI/IEEE-ANS-7-4.3.2-1982, American Nuclear Society, La Grange Park, Illinois, 1982: (1) Sect. 4, (2) Sect. 4.1, (3) Sect. 4.2, (4) Sect. 7, (5) Sect. 4.3, (6) Sect. 5.

Typically, once the system requirements are established, the hardware and software design processes tend to become relatively independent (see Fig. 4.1). In this stage of the design especially, maintaining good communications between the two activities is important.

Experience with large, complex real-time software systems has shown several areas in which problems are likely to occur, and the auditors should be especially alert to them (Straker and Thomas 1983). These areas are (1) startup, reset, and shutdown (of the safety system software); (2) means for detecting and handling errors and exceptions (particularly for time-critical, interrupt, and null data problems); (3) means for incorporating redundancies and modes in which backup devices are switched in and out; and (4) designs of interfaces between the processors and I/O devices and between redundant processors. It is much more efficient, typically, to correct them and other software deficiencies at the design stage than later in the testing stage.

Development of an auditor's checklist for review of the software portion of the safety system design could be based on the following list of questions.

1. Do the software system architectures and basic logic structures used in the design already have proven track records? If not, can the proposed (novel) design be shown (mathematically) to have equivalent or superior reliability attributes to a proven system that could meet the requirements?

2. Has the I/O design interface with the software properly allowed for the dynamic range, accuracy, timing, interrupts, and granularity of the signals?

3. Is the software for the I/O interfaces including interrupts sufficiently robust that it can accommodate, detect, report, and recover from I/O errors? Have verbal communications been eliminated or minimized?

4. Does the overall safety system allow for on-line maintenance, on-line defective I/O module replacement, and on-line module restart?

5. Are the operating system design characteristics suitable, and has the selected operating system had a satisfactory history of operation and debugging for similar applications?

6. Does the software design consider all (reasonably) possible combinations of initialization/startup, operating sequences, and component failure modes, and does it provide for on-line reset (automatic or manual) in case of possible system hangup?

7. Does the timing analysis used in the design allow for adequate margins with some slack for variations in, for example, clock and sampling rate and sensor response time? Has the memory sizing allowed sufficient margin for operations and maintenance and potential upgrades?

8. Does the software design for independent (redundant) safety channels or modules use different programming languages and operating systems for each channel to minimize the chances of common-mode failure?

9. Are the design groups for redundant-channel software kept independent?

10. Is the independence of the software maintenance and upgrades for redundant systems to be maintained throughout the life cycle?

11. Does the design incorporate suitable acceptance test interface capabilities, and are appropriate simulations and other test equipment factored into the design and testing program?

12. Has on-line in-service testing been factored into the design? If so, do the displays associated with on-line testing give clear indications of system performance and error sources?

13. Is a table (requirements matrix) available that shows each system performance and interface requirement and how/where it is met in the software design?

14. Do the analyses of reliability (mean time to failure or mean time between failures) and availability have adequate (supportable) data and sufficient margin to provide reasonable assurance of meeting the requirements?

15. Has the design provided for ease of software (operating system and applications programming) maintenance and upgrades? Has the designer made sufficient allowance for equipment (and perhaps software) obsolescence?

16. Have the algorithms, signal conversion, and data-handling routines used in the applications programs been kept as simple and robust as possible, and do they conform to industry standards?

17. Has the software design met the system requirements for consistency, accuracy, error tolerance, and modularity?

18. Have appropriate human factors considerations been incorporated into the design such that the operator (and maintenance technician or engineer) can readily assess the status of the systems and pinpoint problem areas?

19. Were reactor operators used or consulted in the development of the operator interface system?

Detailed requirements for the software design documentation are described in Sect. 3.2 of the standard on digital safety systems (ANSI/IEEE-ANS-7-4.3.2-1982). In addition, related requirements for integration of the hardware and software are described in Sect. 3.3. All of these requirements should be incorporated in the software design audit checklist.

integration of the hardware and software are described in Sect. 3.3. All of these requirements should be incorporated in the software design audit checklist.

Section 4.2 of ANSI/IEEE-ANS-7-4.3.2-1982 addresses detailed design requirements. After the specific design, safety, and licensibility issues have been resolved, detailed design, coding, and implementation are performed. The detailed design document may follow an outline such as that for the program maintenance manual specified in NIST. The contents, shown in Fig. 4.4, include the definition of all software units (i.e., lower level structures in the software components defined in the architecture), interfaces, intertask messages, and test cases for functional testing.

4.8 ENVIRONMENTAL QUALIFICATION CONSIDERATIONS

Environmental qualification (EQ) considerations are becoming more important because of the increasing concern about low-probability accidents with serious consequences. Many events in this category can be postulated as the result of cataclysmic earthquakes and floods or sabotage and therefore rely on the safety system designs to be much more robust. Although environmental parameters such as temperature, humidity, and vibration directly affect hardware design more than they affect software, the software design is affected in that it should be designed to accommodate disabled sensors, I/O equipment, and even processors themselves. Protecting the system against sabotage and computer viruses is clearly a challenging software design problem. Probably the best defense is to employ verified PROMs in double-locked cabinets.

4.9 ACCEPTANCE TESTING

The three major parts of the acceptance testing phase of the life cycle, which is to demonstrate that the integrated software-hardware system will meet the system requirements, are (1) test plan development, (2) test execution, and (3) analysis of test results. Table 4.3 shows the elements of the audit plan to which these guidelines apply.

The test plan should systematically cover all system requirements and provide metrics for quantitative judgment of system performance. The effectiveness of the tests themselves may be checked by systematically "seeding" the system with errors. Performance in the face of component failures with various failure modes should also be determined.

The major difficulty in the testing process is that it involves a need to ensure correct response to nearly limitless combinations of conditions. Two methods that may be considered to address this problem are (1) "formal methods" with mathematical verification and (2) "statistical methods," which reduce considerably the number of system state combinations that need to be tested. The methodology and implementation used should be evaluated carefully.

1. GENERAL INFORMATION
   1.1 SUMMARY
   1.2 ENVIRONMENT
   1.3 REFERENCES

2. PROGRAM DESCRIPTIONS [2.1 through 2.7 (one for each software unit)]
   2.1 PROBLEM AND SOLUTION METHOD
   2.2 INPUT
   2.3 PROGRAM STRUCTURES AND ALGORITHMS
   2.4 OUTPUT
   2.5 INTERFACES
   2.6 DATA STRUCTURES
   2.7 RUN DESCRIPTIONS

3. OPERATING ENVIRONMENT
   3.1 HARDWARE
   3.2 SUPPORT SOFTWARE
      3.2.1 Distributed Operating System
      3.2.2 Network Operating System
      3.2.3 Compiler
      3.2.4 Debugger
      3.2.5 Other Software

4. MAINTENANCE PROCEDURES
   4.1 PROGRAMMING CONVENTIONS
   4.2 VERIFICATION PROCEDURES
   4.3 ERROR CORRECTION PROCEDURES
   4.4 SPECIAL MAINTENANCE PROCEDURES
   4.5 LISTINGS AND FLOWCHARTS

Fig. 4.4 Example contents of detailed design document for software.

Table 4.3 Validation portion of the elements of the audit plan

Validate computer system requirements
Validate safety system requirements
Examine test plans, procedures, and test reports
Examine metrics and criteria for acceptance of results
Determine applicant/vendor criteria for ending testing
Analysis errors found during testing

Source: Based on data from "Application Criteria for Programmable Digital Computer Systems in Safety Systems of Nuclear Generating Stations," ANSI/IEEE-ANS-7-4-3.2-1982, Sect. 6, American Nuclear Society, La Grange Park, Illinois, 1982.

Development of an auditor's checklist for a review of the software acceptance testing could be based on the following list of questions.

1. Have the test requirements and specifications been shown to provide adequate (complete) coverage of the system requirements? This may best be confirmed by use of a requirements (or capabilities) matrix as described in NSAC (1981). An example of a capabilities matrix, from NSAC (1981), is shown in Fig. 4.4.

2. Have the test procedures and test equipment specifications been clearly defined such that the tests could be repeated (by others) to verify the results?

3. Have the test plan, procedures, and evaluation methodology received an independent (and documented) review?

4. Have the test results been properly archived along with analysis reports, nonconformance documentation, and any corrective actions taken?

4.10 INSTALLATION AND FIELD TESTING

The objective of the audit of the installation and field testing phase of the life cycle is to ensure that the system has been installed properly. The need to repeat any of the previous validation tests probably would depend on whether differences in the field environment (e.g., noisy input signals or power supply lines) or sensor responses (vs those assumed in the design) became apparent. As in the case of the acceptance test audits, those features requiring verification to show compliance with the system and design requirements should be

covered-again, perhaps, by means of the capability matrix. The checklist shown in Sect. 4.7 would also apply here. Table 4.4 lists elements of the audit plan to which these comments apply.

Table 4.4 Installation portion of the elements of the audit plan

Installation and checkout
Verification of installation
Operator's manual

Although the issues of physical control, physical security, configuration control, and labeling procedures of the computer chips should be addressed at an audit prior to installation, it is appropriate to check these procedures on site. This checking can be done by examining plant documents, signature standards, and physical access to storage areas and others. Particular attention should be placed on the means used to physically label the chips and how use of the correct version is ensured (may include, for example, color-coded or keyed sockets, operator-interrogatable burned-in identifications).

4.11 MAINTENANCE AND UPGRADES

Critical reviews and audits of software maintenance and upgrade activities are crucial because, in many cases (after a long time has elapsed), those performing the changes will not be the original design/implementation team. Hence, it is essential that persons making changes understand the bases for the system requirements and the reasoning behind the original software design requirements and implementation. Table 4.5 shows the elements of the audit to which these guidelines apply.

Development of an auditor's checklist for a review of maintenance and upgrade activities could be based on the following questions.

1. Is an adequate, formalized requirements problem reporting system in effect to help ensure that problems will be addressed promptly?

2. Has the maintenance/upgrade (M/U) change procedure been formalized in a way that ensures a systematic and verifiable approach to the modification?

3. Does the change request procedure call for an analysis (when appropriate) of the impact of the change on performance and on an assessment of potential side effects?

4. Does documentation exist to show that M/U changes have had proper approval, documentation, independent V&V, and (where appropriate) in-use tests?

5. Is the M/U change procedure "reasonable" enough that it will not deter making changes that really should be made?

6. Who has responsibility for maintenance and upgrades? (If it is not the vendor, the audit team must examine the applicant's conformance to the provisions of ANSI/IEEE-ANS-7-4.3.2-1982 as closely as that of the vendor.)

7. How is it ensured that the proper version is installed correctly (checksums, physical labeling)?

8. Is the change procedure as controlled as the original implementation (engineering change request, specified signature levels, provisions for repeating V&V)?

Table 4.5 **Maintenance and upgrade portion of the elements of the audit plan**

Post installation procedures
Security
Physical control and identification of computer chips
Software upgrades
Responsibility for upgrades
Procedure for verification and validation of software upgrades
Maintenance
Maintenance manual
Procedures for maintenance