YANKEE NUCLEAR SERVICES DIVISION

FRAMINGHAM, MASSACHUSETTS

Engineering Department Instruction

TITLE: COMPUTER CODES

INSTRUCTION NO: WE-108

REV.	REVIEWS (Initial)				APPROVAL	DATE
	P.E. DEPT.	N.E. DEPT.	E.E. DEPT.	QA DEPT.		
0	DAH.	BCS	JHL	WKP		12/3/84
1						
2						
3						
4						
5						
6						
7						

1.0 PURPOSE

1.1 To provide guidance for development, acquisition, modification, verification, testing, and approval of computer codes that will be used for WE-103 calculations.

2.0 DISCUSSION

2.1 Engineering Instruction WE-103 requires that computer codes used in calculations and analyses have been verified and are appropriate for their applications. This Engineering Instruction (EI) provides the requirements to assure proper verification and documentation for codes developed within YNSD and for those obtained externally, and to assure proper controls for the use of computer codes.

This Engineering Instruction provides instructions and guidance in meeting the procedural requirements. Additional guidance for complex tasks is provided in YAEC-1336, the Systems Development Standards Handbook. This handbook contains very detailed instructions for all aspects of computer code development, testing, and documentation. This Engineering Instruction (EI) provides a summary of these instructions to provide guidance in meeting the procedural requirements. The handbook along with ANS 10.4, Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs (and other referenced guidance documents) should be used if more detailed guidance is required.

2.2 Computer code status categories are defined as follows:

2.2.1 Level I - Computer codes which are in process of being developed, modified, or tested. This includes codes obtained externally which have not been verified internally for a given application.

2.2.2 Level II - Computer codes which have been verified and have documentation to demonstrate verification and to identify the scope or limitations of code usage.

Verification is the act of confirming the methodology (logic) used to calculate or process data is correct per the specifications and requirements stated.

2.2.3 Level III - Computer codes which were previously Level II and then superseded. Documentation for these codes identifies known restrictions and options that are no longer usable. Level III codes may be used in conjunction with their previous usage provided their use is approved by the Project Engineering Manager or Group Manager.

2.2.4 Level IV - Computer codes which are out-of-date, inappropriate, or no longer applicable and are kept for historical purposes. Such codes should be stored in source code format.

2.2.5 Computer codes used prior to January 1, 1984 may be "grandfathered" to Level II or III on the basis of previous satisfactory usage, provided the documentation contains a statement of the basis for usage and verification signed by the Project Engineering Manager or Group Manager, and a description of the scope or applicability of the code. Codes not "grandfathered" shall be assigned a Level I or Level IV status.

2.3 The use of this Engineering Instruction is not required for code verification if an existing code or program can be verified through the normal design analysis review required by the use of WE-103 for calculations and analyses.

3.0 INSTRUCTIONS

3.1 General

3.1.1 The intent of this Engineering Instruction is to assure that computer codes are obtained, prepared, and used in a systematic manner, properly verified, and provided with documentation to demonstrate verification, and allow the scope of applicability to be determined for future usage.

Note: Applicability applies not only to technical scope; even if technically correct, consideration must be given to other restrictions such as plant-specific or non-generic limitations.

3.1.2 The overall process for the development and use of computer codes is shown in Figure 1. The basic requirements are delineated in the following step-by-step instructions. Guidance for accomplishing these requirements is provided in Table 1. The cognizant engineer shall determine the extent to which this guidance is applied for each application depending on safety significance, complexity, degree of standardization, state-of-the-art, and similarity to previous work.

3.2 Initiation (See Table 1)

3.2.1 When it has been determined that there is a need for computer code capabilities presently unavailable on the system as Level II, a Code Cognizant Engineer (CCE) shall be assigned by the Project Engineering Manager or Group Manager.

3.2.2 The CCE shall perform an evaluation in which he will document the problem in a Statement of Problem and describe the requirements that the application must satisfy in a Requirements Specification. From this evaluation the CCE will determine if an existing code should be modified or a new code should be developed.

3.2.3 The CCE shall obtain authorization to acquire a code or to have a code developed or modified in accordance with the steps outlined in Technical Administrative Guide No. 9, "Policy for Control of Computer Hardware and Software Expansion".

3.2.4 If purchase of an existing code is authorized, the CCE shall initiate the purchase of the code in accordance with instructions found in WE-205.

3.2.5 If the CCE is authorized to prepare or modify a code himself, or if a detailed specification is required for acquisition of a code, he shall proceed to Section 3.3.

3.3 Development - Level I (See Table I)

3.3.1 When authorization has been obtained to prepare or modify a computer code, the CCE shall prepare a design specification. He may request assistance from the Computer Services Department (CSD) to the extent he deems necessary for the application.

3.3.2 The specification shall be given a peer review or walkthrough by engineer(s) in the same discipline as the CCE. Reviewers are appointed by the CCE's manager or Lead Engineer.

3.3.3 The CCE shall revise the specification to incorporate (or resolve) the comments of the reviewer(s). The final version of the specification shall be signed by the CCE, reviewer(s), and the CCE's manager or Lead Engineer.

3.3.4 If the specification is to be used for purchase of a computer code, the CCE shall initiate the purchase of the code in accordance with the instructions found in WE-205.

3.3.5 The CCE shall proceed with the design, documentation (see Figure 2), and testing of the code. The CCE may request the services of CSD to the extent he deems necessary for the application, as follows:

(1) Prepare a Standard Memorandum (or Service Request) from the CCE's Department Manager (or Project Manager) to the CSD Manager, providing the scope of services requested and attaching the specification.

(2) Notify the Software Control Library (SCL) of the planned code addition or modification by filling out the appropriate forms according to CSD procedures.

(3) During the development and acceptance of the code, the CCE shall provide technical assistance for CSD as requested, and shall participate in the testing of the code to assure that the requirements of the specification are met.

3.3.6 Final design review and formal approval to Level II shall be as follows:

(1) Final design and testing documentation (see Figure 2 for documentation contents) shall be independently reviewed by an engineer in the same discipline as the CCE using the review guidelines set forth in WE-103 for the review and documentation of review for calculations and analyses.

If the code was obtained from an external source, the CCE shall determine the extent of testing and review required depending on the amount of verification and

validation supplied by the vendor, whether the vendor has an approved quality assurance program, the complexity of the design, and the safety-related status of the code. The CCE shall resolve any verification and validation deficiencies in the vendor's code and documentation. The resolution shall be documented.

(2) The reviewer shall document completion of the review by signing and dating the cover form (see Figure 3).

(3) Upon completion of the review process, the documentation package shall be forwarded to the CCE's manager for approval and signature. Completion of the cover form constitutes final engineering approval and designates a code as certified for Level II usage.

(4) Completion and return to CSD of the appropriate SCL forms designates a code as ready for inclusion in the Software Control Library.

(5) The CCE shall forward the approved documentation package to DCC in accordance with the internal interface and DCC transfer requirements of WE-002.

3.4 Usage - Level II

3.4.1 Codes that have been certified as Level II may be used in accordance with WE-103 for design calculations and analyses or other work requiring quality assurance.

3.4.2 If a code certified for Level II usage replaces or revises an existing code and the CCE determines that continued use of the earlier code is either inadvisable or unwarranted, he shall contact the Software Control Library to request archival status for the application.

The CCE shall coordinate the archival procedure to ensure that there will be no impact on any other group or department.

3.4.3 The CCE shall be responsible for following the usage of those codes which he verified for Level II usage, and he shall be notified according to CSD procedures of any discrepancies in such codes.

(1) Note discrepancies.

(2) Document notification of other users.

(3) Acquire information regarding the impact of these discrepancies.

(4) Assess impact.

(5) Accumulate all records in a program file.

3.4.4 If it is determined that modification is necessary to resolve discrepancies and that future use and availability of the code is desired, the CCE shall implement the change process by returning to the initial instructions of this procedure and following through where applicable.

3.5 Levels III and IV

3.5.1 Codes assigned archival status are defined as either Level III - Controlled Use or Level IV - Archive.

(1) Codes designated as Controlled Use status may be accessed with Group Manager or Project Engineering Manager approval.

(2) Codes designated as Level IV have Archive status.

4.0 RECORDS

4.1 When computer codes are used in calculations and analyses, the records requirements of WE-103 shall apply.

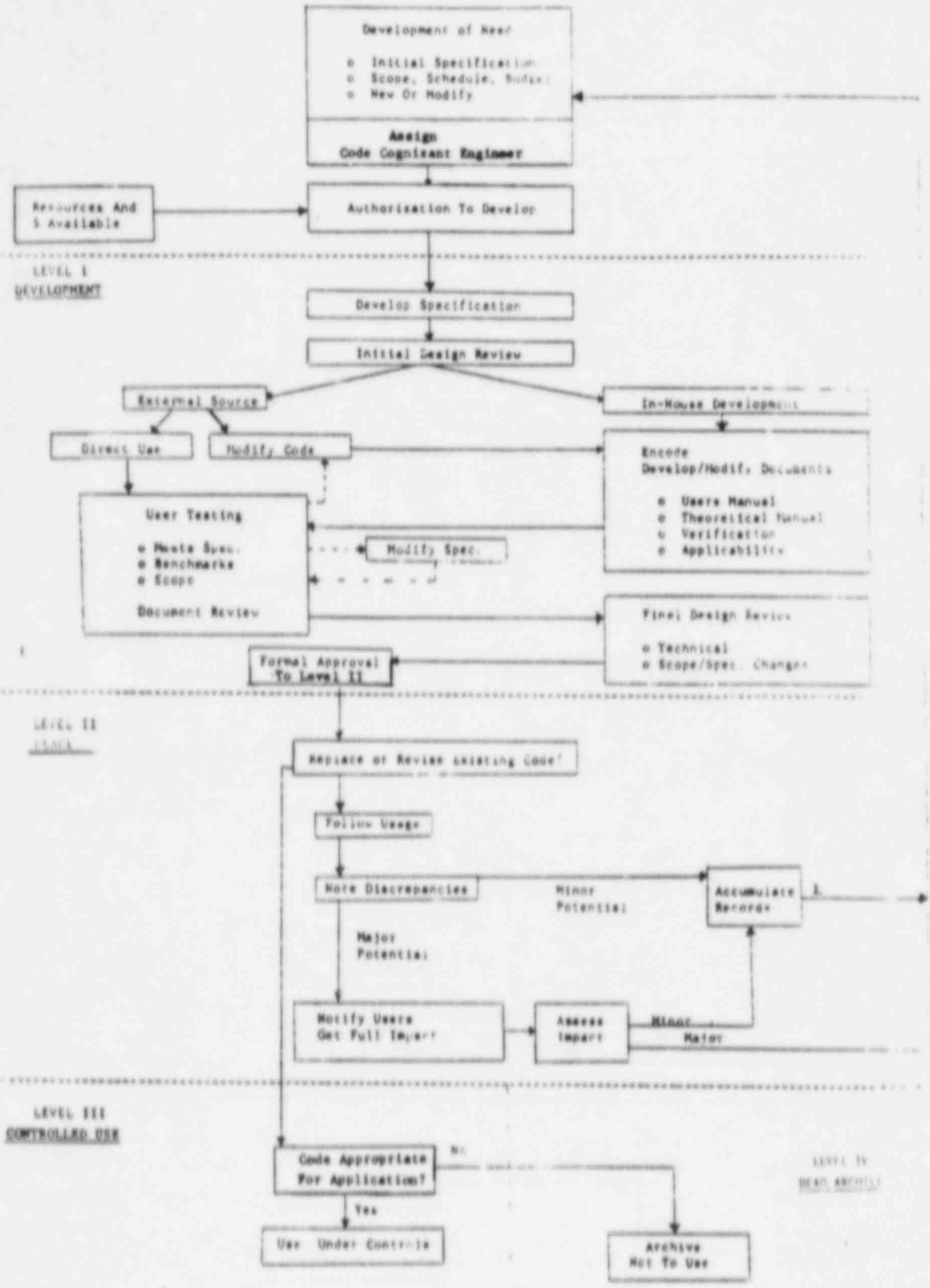

FIGURE 1

WE-108-9

DOCUMENTATION

Purpose/Objective - This is a description of the reasons behind the initial calculation or revision.

Discussion - This is a description of the initial calculation or revision. This should include consideration of scope, applicability, and limitations, including plant-specific acceptability.

Input and Output - This section usually contains the code manual normally provided with the code. As a minimum, will contain an input and output description and requirements.

Programmers Information - This should include a compiled listing of the code on fiche using appropriate update options for cross referencing.

Operational Information - This section should contain hardware requirements, job control language (JCL), and update information as applicable.

Validation - Items a through d below should be repeated for each test case.

a) Description of test case

b) Test case input with JCL

c) Test case output

d) Test case evaluation

Data Retrieval Information - This should include tape number, file numbers, and file names, as applicable.

Software Control Information - This will include the Software Control Library Forms, if applicable.

User Feedback - The intent of this section is to provide a location for user comments if they are accumulated. These comments may be incorporated in following revisions. It is not necessary that this section be reviewed and approved.

FIGURE 2

PAGE 1 OF _____ PAGES

IMS NO. _____

RECORD TYPE _____

W.O./P.O. NO. _____

YANKEE ATOMIC ELECTRIC COMPANY COMPUTER CODE FOR

TITLE _____

PLANT _____ CYCLE _____

NUMBER _____

	PREPARED BY/DATE	REVIEWED BY/DATE	APPROVED BY/DATE
ORIGINAL			
REVISION 1			
REVISION 2			
REVISION 3			

KEYWORDS _____

_____

FIGURE 3

WE-108- 11

TABLE 1

Guidance for the Development and Use of Computer Codes

Requirements of WE-108	Summary Guidance	Detailed Guidance
3.1 GENERAL		
3.1.2 Engineer Uses Guidance	Cognizant engineer determines extent to which guidance applies depending on: Safety significance Complexity Degree of standardization Similarity to previous work	
3.2 INITIATION		
3.2.1 Appoint Code Cognizant Engineer (CCE)		
3.2.2 CCE performs evaluation	Prepare Statement of Problem and Requirements Specification	ANS 10.4
	Analyze current system	SDS Handbook Phase I - Feasibility Study Pages 4 to 42
	Analyze requirements and prepare recommendation	
	Prepare preliminary design for proposed system	
	Establish costs, benefits and schedules	
	Prepare feasibility report	
	Deliverables: o Functional specification o General system design	
3.2.3 CCE obtains authorization	Recommend purchase/modification	TAG No. 9
	Requests involving costs of less than $1,000 require only approval by CSD Manager	
	Request for authorization requires the following steps: Verify similar efforts not underway Determine realistic system life cycle costs Provide notification to CSD before a new program or system is acquired or developed	
	Prepare request to justify recommendation	
	Obtain Department Manager approval	
	Submit request to Management Review Board in the following format: Functional requirements System description Application(s) Features and benefits Procurement costs Support costs Major implementation milestones Potential future expansion	

TABLE 1 (continued)

Guidance for the Development and Use of Computer Codes

Requirements of WE-108	Summary Guidance	Detailed Guidance
3.2.4 If purchase, CCE initiates process for purchase	Prepare supply requisition	WE-205
3.2.5 If modification or preparation, CCE starts development instructions (Level I)	Proceed to Section 3.3	WE-108
3.3 DEVELOPMENT - LEVEL I		
3.3.1 CCE (and/or CSD) prepares specification	Design detailed logical system: What functions will be performed What circumstances the functions will be performed under What data will be acted upon What processing will be required What output will be produced Design detailed physical system Plan the implementation Determine test and operational environment Revise costs, benefits, and schedules Prepare System Specification Report Deliverables: o Design specification o Acceptance test plan o System test plan	SDS Handbook Phase II - Detail System Design Pages 43 to 75
3.3.2 Specification is reviewed	Reviewers assure that the design specification: Completely defines the proposed code or system Will produce a functional code or system at the estimated cost and schedule For guidance in performing specification review see attached tables in WE-108: Table 2 Guidance for Requirements Verification Table 3 Guidance for Verification of Test Plan Table 4 Guidance for Design Verification Table 5 Guidance for Verification of the Program Documentation	WE-108
3.3.3 Specification is revised and approved	Reviewers provide comments to CCE for incorporation or resolution Final version signed by preparer, reviewers, and manager or Lead Engineer	
3.3.4 If purchase, CCE initiates process for purchase	Prepare supply requisition	WE-205

TABLE 1 (continued)

Guidance for the Development and Use of Computer Codes

Requirements of WE-108	Summary Guidance	Detailed Guidance
3.3.5 CCE (and/or CSD) proceeds with encoding, developing/modifying, documenting, and testing of code	Design code and test system Prepare procedures and conduct training Satisfy operations requirements Conduct system test CCE participates in certification of the system test results For guidance in performing system test see attached tables in WE-108: Table 6 Guidance for Source Code Verification Table 7 Guidance for Verification of Program Integration Table 8 Guidance for Program Validation Table 9 Guidance for Verification of Test Results Deliverables: o Tested system/report o User's manual o Training manual	SDS Handbook Phase III - Programming and Procedures Pages 76 to 94
	For format, see Figure 2	WE-108
(1) CCE requests CSD services	Prepare standard memorandum (or Service Request) and attach specification	WE-005
(2) CCE notifies Software Control Library of planned addition or modification of code.	Prepare appropriate forms: o SCL Support Request o Product Installation Request (if initial installation of code) o Modification Request (if modification of previously installed code) o Test Results Report Note: Signatures of CCE and reviewer are not entered until final testing is complete o Product Installation Release	CSD Procedures SCL Forms
(3) CCE assists in development and acceptance process training	Complete procedures and conduct formal training Conduct acceptance test of operational system CCE participates in certification of system test results Deliverables: o Operational system	SDS Handbook Phase IV-System Acceptanc Pages 95-99

TABLE 1 (continued)

Guidance for the Development and Use of Computer Codes

Requirements of WE-108	Summary Guidance	Detailed Guidance
3.3.6 Final design review and formal approval to Level II		
(1) Independent review by engineer in same discipline as CCE	Use reviewing guidelines delineated in WE-103	WE-103
	For documentation contents, see Figure 2	WE-108
If code obtained externally, CCE determines extent of testing required	Depth of review dependent on: Amounts of verification and validation supplied by vendor Whether vendor has an approved quality assurance program Complexity of design Safety-related status of code	
	CCE resolves deficiencies in vendor documentation and documents resolution	
(2) Reviewer documents completion of review	Signs and dates form, Figure 3	WE-108
(3) Certification of code for Level II usage	CCE's manager approves documentation package. Signs and dates form, Figure 3	
(4) Designation of code as ready for inclusion in SCL	Appropriate forms completed	CSD Procedures SCL Forms
(5) Documentation package transferred to DCC	Use transfer steps delineated in WE-002	WE-002
3.4 USAGE - LEVEL II		
3.4.1 Code may be used for design analyses or work requiring QA approval	Use reviewing guidelines delineated in WE-103 o Verify that code is Level II o Verify that code is appropriate	WE-103
3.4.2 Existing code is replaced or revised	CCE initiates archival process	CSD Procedures
CCE decides continued used of code is inadvisable	Contact Software Control Library Coordinate archival procedure	
3.4.3 CCE follows usage	Update procedures and conduct training at follow-on locations Provide post-installation support Prepare for post-installation audit Install operational system at follow-on locations	SDS Handbook Phase V - Implementation and Suppo Pages 100-104
	For guidance in performing installation, see attached table in WE-108: Table 10 Guidance for Verification of Installation Package	WE-108

TABLE 1 (continued)

Guidance for the Development and Use of Computer Codes

Requirements of WE-108	Summary Guidance	Detailed Guidance
3.4.3 (continued) - Users shall notify CCE of discrepancies	Note discrepancies	CSD Procedures
	Document notification of other users	
	Get full impact	
	Assess impact	
	Accumulate records in program file	
3.4.4 CCE identifies need to modify	Modification is required:	WE-108
	Return to beginning of Procedure WE-108 and follow where applicable	
3.5 ARCHIVAL STATUS - LEVELS III AND IV		
(1) Controlled Use	Used under control	
	Available for use with Group Manager or Project Engineering Manager approval	
(2) Archive	Archive status	
4.0 RECORDS		
4.1 Control Requirements	For codes used in calculations and analyses, use records requirements delineated in WE-103	WE-103

TABLE 2

Guidance for Requirements Verification

1. Does the Requirement Specification conform to required documentation standards?

   a. Are all required sections present?

   b. Does each section contain all of the required information?

   c. Is the format as required?

2. Does the Requirement Specification reflect an understanding of the problem to be solved?

   a. Are the requirements consistent with the Statement of Problem for the program?

   b. Are the models that are specified appropriate for the problem to be solved?

   c. Are the numerical techniques that are specified appropriate for the problem to be solved?

   d. Are the algorithms that are specified appropriate for the problem to be solved?

   e. Have program functions been partitioned in a manner consistent with the problem to be solved?

   f. Will the program as specified solve the problem?

3. Are the requirements complete?

   a. Do the requirements include all functions called for or implied by the Statement of Problem?

   b. Do the requirements identify all program interfaces and fully specify required program behavior with respect to each?

   c. Are all program inputs identified and described to the extent needed to design the program?

   d. Are all required program outputs identified and described to the extent needed to design the program?

   e. Does the specification describe the operational environment into which the program must fit?

   f. Does the specification include all desired quality requirements, e.g., requirements for performance, reliability, accuracy, portability, maintainability, user friendliness?

   g. Does the specification reference all desired development standards?

   h. Does the specification include acceptance criteria for the program?

   i. Does the specification include timing and sizing constraints if applicable?

   j. Does the specification include required behavior in the face of improper inputs and other anomalous conditions?

4. Are the requirements correct?

   a. Are all requirements consistent with the Statement of the Problem for the program?

   b. Are all requirements consistent with documented descriptions and known properties of the operational environment into which the program must fit?

TABLE 2 (continued)

Guidance for Requirements Verification

c. Do interface requirements agree with documented descriptions and known properties of the interfacing elements?

d. Do input requirements correctly describe all inputs whose format, content, data rate, etc., are not at the discretion of the designer?

e. Do output requirements correctly describe all outputs whose format, content, data rate, etc., are not at the discretion of the designer?

f. Do requirements concerning models, algorithms, and numerical techniques agree with standard references, where applicable?

5. Are the requirements consistent?

a. Is the Requirement Specification free of internal contradictions?

b. Are the models, algorithms, and numerical techniques that are specified mathematically compatible?

c. Are input and output formats consistent to the extent possible?

d. Are the requirements for similar or related functions consistent?

e. Are the accuracies required of inputs, computations, output, etc., compatible?

f. Are the style of presentation and the level of detail consistent throughout the document?

6. Are the requirements clear and unambiguous?

a. Can all requirements be interpreted?

b. Can all requirements be interpreted in only one way?

c. Are the requirements organized and presented in a way that promotes clarity (for example, use of tables and lists in place of text, where applicable)?

d. Are the requirements sufficiently detailed to prevent misinterpretation?

e. Does the specification differentiate between program requirements and other information provided in the specification?

7. Are the requirements feasible?

a. Are the specified models, algorithms, and numerical techniques within the state of the art?

b. Can they be implemented within the constraints imposed on the system and on the development effort?

c. Are the quality attributes specified for the program achievable both individually and as a group?

d. Are the required functions attainable within the available resources?

8. Does the Requirement Specification contain adequate provision for program validation and acceptance?

a. Is each requirement testable?

b. Are acceptance criteria specified?

TABLE 2 (continued)

Guidance for Requirements Verification

c. Are the acceptance criteria consistent with:

- Results obtained from similar computer programs
- Classical solutions
- Accepted experimental results
- Analytical results published in technical literature
- Benchmark problems

9. Does the Requirement Specification avoid placing undue constraints on program design and implementation?

a. Does the specification avoid telling how the program is to be designed or implemented?

b. If it does place constraints on design or implementation, is there justification for including such constraints in the requirements?

TABLE 3

Guidance for Verification of the Test Plan

1. Is the Test Plan description complete?

   a. Is the software product to be tested fully identified?

   b. Are the scope and objectives of the Test Plan identified?

   c. Are the scope and objectives consistent with the Requirement Specification?

   d. Are the Requirement Specification and other documents required for Test Plan development and execution referenced?

2. Is the testing approach to be followed described?

   a. Are requirements to be tested identified?

   b. Acceptance criteria defined?

   c. Are methods used for indicating compliance identified?

3. Are the test problems' definitions adequate?

   a. Is the basis for selection specified?

   b. Is the description of test programs complete?

   c. For complex applications, is there at least one problem that will provide unequivocal analysis of results with minimum human judgement?

   d. Does each problem have known and accepted results?

   e. Are the numerical accuracy limits of each problem consistent with the software product?

   f. Is the application range of the software product, as defined in the Requirement Specification, adequately covered by the set of test problems?

   g. Are all types of problems that are identified for solution by the software included in the set of test problems or excluded for a specified, valid reason?

4. Is each testable requirement adequately covered?

   a. Is at least one test case proved for each requirement?

   b. If the requirement covers a range of values or capabilities, are test cases identified to cover the range adequately?

   c. Is the rationale for selecting test cases clear and valid?

   d. Does the testing methodology establish an unambiguous means of determining that the program complies with specified requirements?

   e. Is the matrix of test cases versus requirements complete?

   f. Are redundant test cases avoided?

5. Are the test case descriptions completed?

   a. Is each test case derived from a documented testing approach?

   b. Are the test cases consistent with the documented testing approach?

   c. Does the test case matrix clearly establish the relationship between test cases and requirements being tested?

TABLE 3 (continued)

Guidance for Verification of the Test Plan

d. Is the specification for each test case complete?

- Unique identification
- Objective(s)
- Input
- Expected results
- Evaluation criteria
- Relation to other tests
- Rationale for test setup

e. Are there test cases specified that are representative of conditions under which the program will be utilized?

6. Is the specification for each test case adequate?

a. Is input detail sufficient for input preparation?

b. Are expected results explicit and specified with sufficient accuracy?

c. Do the evaluation criteria provide unambiguous pass/fail status for each test case?

d. Is the method of comparing test output with expected results feasible?

e. Are dependencies between test cases described and adequately specified?

f. Are data libraries identified?

g. Are the operating environment requirements feasible?

7. Is the plan for building, updating, and maintaining the test data base adequate?

a. Is every test case represented?

b. Does the input agree with test case specifications?

c. Are output options correctly specified?

d. Is all output required for evaluation of test case results specified?

e. Are full references and all job control language consistent with specifications?

f. Is each job stream executable?

g. Is each input case correctly identified?

h. Are input preparation instructions in accord with the program documentation?

8. Are the test procedures complete?

a. Are the procedures specified in sufficient detail to permit third-party execution?

b. Do the procedures cover building, updating, and use of the list data base?

c. Will all required test results be produced?

d. Are instructions provided for disposition of test results?

e. Do the procedures cover preparation and use of all necessary data files and external support programs?

f. Do the procedures provide for configuration control of the test data base, data files, and external programs, consistent with the V&V plan?

TABLE 3 (continued)

Guidance for Verification of the Test Plan

g. Are the procedures automated wherever possible to minimize human error?

h. Is the evaluation methodology for each test case described?

i. Is the use of discrepancy report forms specified in the procedures?

j. Is a procedure for rerunning test cases included?

9. Can the procedures be performed within the planned resources?

10. Is the plan for reporting test results adequate?

a. Is there a recommended format for discrepancy reports?

b. Is the test case log format specified?

c. Is the information required on the test case log sufficient to document date and time of executions, observations that failures (discrepancies) have occurred, and responsibility for test execution?

d. Is an adequate method of summarizing test results provided?

e. Will the test results provide the input needed to demonstrate that the program meets its acceptance criteria?

f. Are there procedures to control retesting and document change?

g. Is the method for assembling individual test results into the Test Report described?

TABLE 4

Guidance for Design Verification

1. Does the Design Specification conform to required documentation standards?
   a. Are all required sections present?
   b. Does each section contain all of the required information?
   c. Is the format as required?
2. Is the Design Specification traceable to the Requirement Specification?
   a. Are all requirements implemented in the design?
   b. Do all aspects of the design have their basis in the requirements?
   c. Are the numerical techniques that are specified appropriate for the problem to be solved?
   d. Are the algorithms that are specified appropriate for the problem to be solved?
   e. Has the program design been partitioned in a manner consistent with the problem to be solved?
   f. Will the program as designed meet the requirements?
3. Is the design complete?
   a. Does the design implement required program behavior with respect to each program interface?
   b. Are all program inputs, outputs, and data base elements identified and described to the extent needed to code the program?
   c. Does the specification describe the operational environment into which the program must fit?
   d. Are all necessary processing steps included?
   e. Are all possible outcomes of each decision point designed?
   f. Does the design take into account all expected situations and conditions?
   g. Does the design specify appropriate behavior in the face of unexpected or improper inputs and other anomalous conditions?
   h. Does the specification reference all desired programming standards?
4. Is the design correct?
   a. Is the design logic sound, that is, will the program do what is intended?
   b. Is the design consistent with documented descriptions and known properties of the operational environment into which the program must fit?
   c. Do interface designs agree with documented descriptions and known properties of the interfacing elements?
   d. Does the document correctly describe all inputs, outputs, and data base elements whose format, content, data rate, etc., are not at the discretion of the designer?
   e. Do the models, algorithms, and numerical techniques used in the design agree with standard references, where applicable?

TABLE 4 (continued)

Guidance for Design Verification

5. Is the design internally consistent?

   a. Is the document free of internal contradictions?

   b. Are the models, algorithms, and numerical techniques that are specified mathematically compatible?

   c. Are input and output formats consistent to the extent possible?

   d. Are the designs for similar or related functions consistent?

   e. Are the accuracies and units of inputs, data base elements, and outputs that are used together in computations or logical decisions compatible?

   f. Are the style of presentation and the level of detail consistent throughout the document?

6. Is the design clear and unambiguous?

   a. Can all design information be interpreted?

   b. Can all design information be interpreted in only one way?

   c. Is the design organized and presented in a way that promotes clarity (for example, use of tables and lists in place of text, where applicable)?

   d. Is the design sufficiently detailed to prevent misinterpretation?

   e. Does the specification differentiate between program design and other information provided in the specification?

7. Is the design feasible?

   a. Are the specified models, algorithms, and numerical techniques within the state of the art?

   b. Can they be implemented within the constraints imposed on the system and on the development effort?

   c. Are the functions as designed implementable within the available resources?

TABLE 5

Guidance for Verification of the Program Documentation

1. Does the Program Documentation conform to ANSI N413-1974 and/or other documentation standards imposed on the document?

   a. Are all required sections present?

   b. Does each section contain all required information?

   c. Is the format as required?

2. Is the information in the Program Documentation consistent with that in the Requirement Specification?

3. Is the information in the Program Documentation consistent with that in the Design Specification?

*4. Is the information in the Program Documentation an accurate reflection of the coded program?

5. Is the description of user input adequate for test planning?

   a. Are all inputs specified?

   b. Are formats fully specified?

   c. Are valid ranges of input values specified?

   d. Are valid data rates specified, if applicable?

   e. Are valid input sequences specified as applicable?

6. Is the information in the Program Documentation internally consistent?

   a. Is the document free of internal contradictions?

   b. Are the style of presentation and the level of detail consistent throughout the document?

7. Is the information in the Program Documentation clear and unambiguous?

   a. Can all of the information be interpreted?

   b. Can all of the information be interpreted in only one way?

   c. Is the information organized and presented in a way that promotes clarity?

   d. Is the information sufficiently detailed to prevent misinterpretation?

*Not applicable in the design phase

TABLE 6

Guidance for Source Code Verification

1. Does the coding conform to specified standards and procedures?

   a. Does the coding conform to ANSI Standard ANS 10.2 on Programming Practices?

   b. Does the coding conform to ANSI standards on the coding language, if applicable?

   c. Does the coding conform to other specific development standards?

2. Are sufficient comment statements provided to give an adequate description of each routine?

   a. Are input and output variables correctly described?

   b. Are constants used in the subroutine described?

   c. Are the various calculations and tasks explained?

   d. Are the reading and writing of I/O files clearly explained?

   e. Are special coding features such as mixed mode, word packing, and non-ANSI coding clearly identified and explained?

3. Is the coding clearly understandable?

   a. Are complex coding structures avoided?

   b. Are consistent indenting and related techniques used to enhance clarity?

   c. Is self-modifying code avoided?

4. Is the source code logically consistent with the Design Specification?

   a. Are all features of the design fully and correctly implemented in the code?

   b. Do all features of the coded program have their basis in the Design Specification?

5. Are all variables properly specified and used?

   a. Is the program free of unused variables?
   b. Are all variables initialized?
   c. Are array subscripts consistent?
   d. Are loop variables within bounds?
   e. Are constants correctly specified?
   f. Are proper units used with each variable?

6. Is there satisfactory error checking?

   a. Are input items checked for limits?

   b. Are external data files checked to assure that the correct data file is being read and the data is in proper format?

   c. Are results of calculations checked to be reasonable values?

   d. Does error detection result in consistent error messages and recovery?

7. Do all subroutine calls transfer data variables correctly?

   a. Are the number of variables and the type of each variable the same in both the calling and called routines?

   b. Are labeled common variable names, type, location, and size of arrays consistent throughout the problem?

TABLE 6 (continued)

Guidance for Source Code Verification

8. Is the data read from each file consistent with the data written to that file?

   a. Are the number and type of variables consistent?

   b. Are unit numbers consistent?

9. Do unit test results show that:

   a. Each routine tested for major logic paths within the routine?

   b. Each routine was checked for appropriate minimum, maximum, and average sets of variables?

   c. Edit statements were used to print intermediate results?

   d. The module reproduces identical results with identical input?

TABLE 7

Guidance for Verification of Program Integration

1. Does the integrated program conform to the system resource requirements?

   a. Does the program meet storage requirements for small core, extended core disk, and tape?

   b. Does the program meet time and sizing requirements?

2. Does the integrated program interface properly with external files?

   a. Does the program properly read and write external files?

   b. Does the program properly read user input data?

3. Are all pieces of the integrated program properly identified?

   a. Has the source code been verified?
   b. Has the compiler been identified?
   c. Have special user libraries been verified?
   d. Have system libraries been identified?

   e. Has the loader been identified?

4. Does the program link together in a consistent manner?

   a. Are there any missing subroutines?

   b. Does the module linkage specification create a properly linked program?

   c. Are all routines loaded into proper segments?

   d. Are global and local labeled common blocks properly specified for each segment?

5. Are the interfaces between functional units correct?

   a. Are labeled commons consistent?
   b. Are argument lists passed consistently?
   c. Are I/O data file names consistent?
   d. Are I/O structures consistent and correct?

6. Is the control language for building the integrated program correct?

   a. Are proper compiler options used?

   b. Are proper libraries specified?

   c. Are loading options consistent for initialization of variables and obtaining load maps?

7. Is the control language used for execution proper?

   a. Are all files properly specified?

   b. Are execution time limits correct?

   c. Are external data files properly attached and saved?

8. Are the special data libraries that are used for execution correct?

   a. Do the libraries conform to the Design Specification in structure and format?

   b. Are there sufficient data in the libraries for proper execution?

   c. Can additional data be added to the libraries easily?

TABLE 7 (continued)

Guidance for Verification of Program Integration

9. Have sufficient edits been produced to verify the processing of data and transmission of data between and among modules?

   a. Have principal I/O files been checked?

   b. Have labeled common blocks been checked?

   c. Have variables passed to routines been checked?

   d. Have principal results been checked?

TABLE 8

Guidance for Program Validation

1. Has each requirement been tested adequately?

   a. Does the set of test results corresponding to each requirement cover the range adequately?

   b. Has each test result for this requirement satisfied its evaluation criteria?

   c. Does the combination of test case results for this requirement meet the acceptance criteria?

2. Is the total set of requirements met?

   a. Is every acceptance criterion met satisfactorily?

   b. Are there any test results that indicate unrepeatable, unreliable or unexpected program behavior?

   c. Are the test results consistent with the initial Statement of Problem for the program?

TABLE 9

Guidance for Verification of Test Results

1. Does the Test Report comply with the format specified in the Test Plan?

   a. Does it provide complete identification of the program tested?

   b. Does it specify the scope of the Test Report?

   c. Does it reference the Test Plan and any other relevant documents?

   d. Does it provide a complete and accurate description of the test environment:

      - Location
      - Equipment
      - Support software used

   e. Does it describe and justify each deviation from the Test Plan?

   f. Does it provide a summary of test results?

   g. Does it provide an evaluation of the program?

   h. Does it provide recommendations for retesting and/or program acceptance?

   i. Does it provide a detailed description of the results of each test case?

   j. Does it include a copy of the test case log?

   k. Does it include all discrepancy reports prepared during the testing?

2. Is the information in the Test Report an accurate reflection of the testing performed?

   a. Does the summary of test results accurately reflect the test outputs?

   b. Is the evaluation of the program a realistic and accurate reflection of the test results?

   c. Are the recommendations regarding retesting and/or acceptance sound and based on the test results?

   d Do the descriptions of test results accurately reflect actual test outputs?

   e. Is the test case log complete and consistent with actual test outputs?

   f. Are the discrepancy reports complete and consistent with actual test outputs?

3. Have all test cases been executed correctly?

   a. Does the test case log indicate performance of each test case in the specified environment, using specified test procedures?

   b. Is there an explanation for any deviation from the specified test environment or procedures?

   c. Is there a discrepancy report for each deviation from expected results?

   d. Were correct inputs used for each test case?

   e. Are the output values for each test case accurately reported?

TABLE 10

Guidance for Verification of Installation Package

1. Are sufficient materials available on the program installation tape to permit rebuilding of the installed program?

   a. Are the necessary pieces from the following list available?

      - Source code
      - User-supplied library routines
      - Systems library routines
      - Module linkage specifications
      - Data structure definitions and data base materials
      - Control language for installation
      - Data libraries to be used by the program
      - Test cases
      - Control language for execution
      - Output results from the test cases

   b. Are the format and content of the tape properly identified in the installation procedures for easy reading of the files?

   c. Are the installation procedures clearly understandable to allow installation and checkout?

2. Can the program be rebuilt from the installation package?

   a. Can the program source be recompiled in the same manner as before?

   b. Can the program be reloaded into an executable program in the same manner as before?

   c. If there are changes in rebuilding, do these changes affect the functional operation of the program?

3. Do the test cases produce results identical to output supplied with the installation package?

   a. Can all test cases be performed?

   b. Are all results identical to previous results?

   c. Are differences in results clearly understood and justified (such as new date and time on printed output)?