Enclosure 6

MFN 07-384

ESBWR Licensing Topical Report –

Software Quality Assurance Plan, Rev. 2 –

NEDO-33245

Non-Proprietary Version

GE Energy
Nuclear

3901 Castle Hayne Rd

Wilmington, NC 28401

NEDO-33245

Revision 2

Class I

DRF#0000-0049-7144

June 2007

LICENSING TOPICAL REPORT

ESBWR - I&C SOFTWARE QUALITY ASSURANCE PLAN (SQAP)

PROPRIETARY INFORMATION NOTICE

This is a non-proprietary version of the document NEDO-33245, Rev 1, and thus, has the proprietary information removed. Portions of this document that have been removed are indicated by open and closed double brackets, as shown here [[ ]].

Important Notice Regarding Contents of this Report
Please read carefully

The information contained in this document is furnished **for the purpose of supporting the NRC review of the certification of the ESBWR**. The only undertakings of General Electric Company with respect to information in this document are contained in contracts between General Electric Company and participating utilities, and nothing contained in this document shall be construed as changing those contracts. The use of this information by anyone other than that for which it is intended is not authorized; and with respect to any unauthorized use, General Electric Company makes no representation or warranty, and assumes no liability as to the completeness, accuracy, or usefulness of the information contained in this document.

]]

[[

]]

Figure 3. Example of a Traceability Matrix Structure 104

[[

]]

1.0 INTRODUCTION

1.1 Overview

The ESBWR Man-Machine Interface System and Human Factors Engineering Implementation Plan (MMIS/HFE IP) [2.1(1)] requires an ESBWR – I&C Software Quality Assurance Plan (SQAP) to be prepared. The SQAP shall describe the Software Quality Assurance (SQA) activities to be performed during the software life cycle phases of the ESBWR Quality Class Q and Quality Class N digital computer-based I&C system, herein referred to as software product.

The SQAP meets the acceptance criteria specified in Chapter 7 of NUREG 0800, Standard Review Plan (SRP) [2.2.1 (1)], except where specified in Appendix A.

1.2 Purpose and Scope

The purpose of the SQAP is to:

1. Establish a SQA program to monitor the software life cycle activities of the software products and to identify the organization responsible for the SQA program and its organizational boundaries.
2. Supplement GE Nuclear Energy (GEEN) Quality Assurance Program, which is in full compliance with 10CFR 50, Appendix B, Quality Assurance Criteria for Nuclear Power Plant and Fuel Processing Plants [2.2.2(1)].

The objectives of the SQA program are to ensure that:

1. The design teams follow:
   - The established GE Policies and Procedures (P&Ps).
   - The Engineering Operating Procedures (EOPs).
   - The requirements described in this SQAP.
   - The ESBWR I&C Software Management Plan (SMP) [2.3(1)] (herein referred to as SMP).
2. The design documentation and design outputs for each software life cycle phase defined in the SMP [2.3(1)] are adequate (i.e., correct and complete).
3. The final software products are acceptable to be installed and ready for operation in a nuclear power plant.

The SQAP defines the SQA activities, methods, and tools by which to execute these activities. The SQAP also specifies the following:

- Required verification and validation (V&V) activities [Section 7.0, Software V&V Plan (SVVP)].
- Software safety analysis (SSA) [(Section 9.0 Software Safety Plan (SSP)]
- Software configuration management (SCM) [Section 10.0 Software Configuration Management Plan (SCMP)]

This SQAP shall be in force during all phases of the software life cycle.

The applicable Software Products (software and firmware) covered by this SQAP encompass all I&C systems, as specifically defined in the MMIS/HFE IP [2.1(1)] (Subsection 1.2.4 only), which perform the monitoring, control, and protection functions associated with all modes of ESBWR plant normal operation (i.e., startup, shutdown, standby, power operation, and refueling) as well as off-normal, emergency, and accident conditions.

1.3 Acronyms, Abbreviations and Definitions

Acronyms and abbreviations are defined in Appendix B.

Definitions are provided in Appendix C.

1.4 Software Developed by Vendors

Software Products developed by the vendors shall comply with this SQAP. If a vendor elects to follow its established SQA program, then the SQA program as defined in the contract/purchase order (Section 14.0, Vendor and Acquired Software Control) shall be reviewed and approved by the SQA Manager to assure compliance with the requirements specified in this SQAP.

1.5 Software Classification

SSA shall be conducted on the software requirements for Quality Class Q and Quality Class N software products to assign the appropriate Software Classification as described in the Table 1.5-1.

If the software performs functions which are classified per EOP 65-2.10 as Safety-Related then it shall be classified as Software Class "Q".

All other software shall be considered Nonsafety and will be divided into two sub-classes "N3", and "N2". A criticality analysis shall be conducted for all Nonsafety software. If there is a failure mode, which could challenge safety systems as defined below, then the software shall be classified as "N3".

- Software whose inadvertent response to stimuli, failure to respond when required, or response out-of-sequence could directly result in a accident or transient as defined in the Document Control Document, chapter 15 [2.1(6)].
- Software that is intended to mitigate the result of an accident.
- Software that is intended to recover from the result of an accident.

The remaining Software shall be classified as "N2". This software is Nonsafety-related system software whose failure cannot adversely affect a safety-related function.

The Software Classification is determined as shown in Figure 1 and is performed during the SSA Preparation phase as described in Subsection 9.3.1. This scheme is based on IEEE Std. 1012 "IEEE Standard for Verification and Validation Plans" [2.2.4(1)].

Table 1.5-1 Software Classification

Classification	Description
Software Class Q	Software performs functions classified per EOP 65-2.10 [2.2.5(2aa)] as Safety-Related.
Software Class N3	Nonsafety-related systems software whose failure could challenge safety systems as defined below: a. Software whose inadvertent response to stimuli, failure to respond when required, response out-of-sequence could directly result in an accident or transient as defined in the Design Central Document, chapter 15 [2.1(6)]. b. Software that is intended to mitigate the result of an accident. c. Software that is intended to support recovery from the result of an accident.
Software Class N2	• Software failure cannot adversely affect a safety-related function. • Software failure results in inconvenience to the user.

[[

]]

2.0 APPLICABLE DOCUMENTS

Applicable documents include supporting documents, supplemental documents, codes and standards and are given in this section. Supporting documents provide the input requirements to this plan. Supplemental documents are used in conjunction with this plan.

2.1 Supporting Documents

The following supporting documents were used as the controlling input documents in the production of this plan. These documents form the design basis for the activities stated in this plan. This document governs, in the event of any differences noted between the SQAP and the ESBWR Composite Design Specification [2.1(2)].

1. ESBWR Man-Machine Interface System and HFE Implementation Plan (MMIS/HFE IP), NEDO-33217, Rev. 2.
2. ESBWR Composite Design Specification (A11-5299), 26A6007, Rev. 0.
3. ESBWR Composite Design Specification "Standard Review Plans and Regulatory Guides" (A11-5299), 26A6007AB, Rev. 3.
4. ESBWR Composite Design Specification Industry Codes and Standards (A11-5299), 26A6007AC, Rev. 2.
5. ESBWR DCD, Chapter 7, I&C Systems, 26A6642AW, Rev. 3.
6. ESBWR DCD, Chapter 15, Safety Analysis, 26A6642BP, Rev. 3.

2.2 Codes and Standards

The following codes and standards are used in conjunction with this plan.

2.2.1 NUREG

The following codes and standards are applicable to the activities specified within this plan. This Plan conforms to planning requirements of these codes and standards except as explicitly noted in Appendix A.

1. NUREG 0800, Standard Review Plan (SRP), Chapter 7, Branch Technical Position (BTP) HICB-14 R4, Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems.

2.2.2 CODE OF FEDERAL REGULATIONS (CFR)

1. 10CFR50, Appendix – B, Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants.

2.2.3 U.S. NUCLEAR REGULATORY COMMISSION (NRC) REGULATORY GUIDES (RG)

The following codes and standards are applicable to the activities specified within this plan. This Plan conforms to planning requirements of these codes and standards except as explicitly noted in Appendix A.

1. RG 1.168-2004 – Verification, Validation, Reviews, and Audits For Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
2. RG 1.169-1997 – Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
3. RG 1.170-1997- Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
4. RG 1.171-1997 – Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
5. RG 1.172-1997 – Software Requirements Specifications for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
6. RG 1.173-1997 – Developing Software Life cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
7. RG 1.152-2006 – Criteria for Use of Computers in Safety Systems of Nuclear Power Plants.

2.2.4 INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS (IEEE)

The following codes and standards are applicable to the activities specified within this plan. This plan conforms to planning requirements of these codes and standards except as explicitly noted in Appendix A.

Where these IEEE Standards provide recommended implementation techniques and methods, this plan makes specific commitments only to those requirements restated herein. The ESBWR Project Work Plans shall capture the detailed implementation attributes in accordance with EOP 25-5.00 [2.3(2a)]. Future exceptions or deviations from the recommendations specified in the IEEE standards shall require management approval as defined in the SMP [2.3(1)] and this SQAP, and are potentially subject to NRC notification. The NRC notification process is addressed in the MMIS/HFE Implementation Plan [2.1(1)].

1. IEEE 1012-1998 Standard for Software Verification and Validation.
2. IEEE 1028-1997 Standard for Software Reviews.
3. IEEE 828-1990 Standard for Software Configuration Management Plans.
4. IEEE-829-1983 Standard for Software Test Documentation.
5. IEEE-1042-1987 Guide to Software Configuration Management.

2.3 Supplemental Documents

The following supplemental documents are used in conjunction with the SQAP and enable the performance of the activities stated in Appendix A. These documents are subject to revision to remain current with GEEN internal procedures, and do not require the SQAP to be updated when they are revised.

1. ESBWR I&C Software Management Plan	NEDO-33226
[[	

	]]
4. IEEE Standard Glossary of Software Engineering Terminology	IEE 610.12-1990
5. Guidelines on Evaluation and Acceptance of Commercial Grade Digital Equipment in Nuclear Safety Applications	EPRI TR-106439

2.4 Additional IEEE Standard Guidance

The following IEEE Standard provides additional guidance for the implementation activities. Conformance of this plan to these activities has been evaluated. Selected sections/topics from this IEEE Standard are excluded from commitment because they either provide conflicting requirements with other Standards or the level of detail is not appropriate for this plan. Clarifications and justifications for such exclusions are provided in Appendix A.

1. IEEE 1228-1994 - IEEE Standard for Software Safety Plans.

2.5 International Standards

1. ISO 9001:2000, Quality Management Systems – Requirements

3.0 MANAGEMENT ORGANIZATION

3.1 Organization

This section defines the functional responsibilities and authorities of the organizations within ESBWR Project who are responsible for the quality of the software products. The organization of the ESBWR Project is shown in Figure 2.

The Quality organization is responsible for GEEN Quality Assurance (QA) program. The Quality Organization is a managerially and financially independent organization. The Quality Manager, who reports to the President and CEO of GEEN, provides leadership for developing and overall coordination of the QA program objectives, including the software quality assurance program. The SQA organization has the overall responsibility for developing and maintaining the SQA program with support from the Software Project Engineering (SPE) organization. The SPE organization is responsible for executing the technical aspect of the software quality assurance program, which includes the following software quality assurance tasks (herein referred to as Quality tasks):

1. Independent verification and validation (IV&V) of Software Class Q software.
2. Software safety analysis.
3. Software Configuration Management (SCM).

The SPE organization is technically, managerially and financially independent from the software products design organization, in conformance with RG 1.168 [2.2.3(1)].

3.2 Activities

The following activities are performed throughout the software life cycle phases:

1. Verification and validation of design documentation and outputs specified in the SMP [2.3(1)].
2. Software safety analysis of Software Class Q software and Software Class N3 software requirements.
3. Software and system testing.
4. Baseline reviews.
5. Software Configuration Control.
6. Software audit.

Table 2 outlines the tasks and the individual or group responsible for conducting these tasks during the design and development of the software products.

3.3 Qualification and Responsibilities

The SQA Manager shall be knowledgeable in the industry standard QA methodologies, and proficient in establishing, maintaining and improving QA Procedures and experience in technical

project management. The SPE members shall be knowledgeable in the technologies and methods used in design development and are qualified to perform the specific software quality assurance tasks. [[

]]

The level of software quality assurance support varies during each software life cycle phase; thus, the membership to the SQA and SPE fluctuate with the level of needs. If necessary, the SQA and SPE Manager has the authority to contract third party organizations (i.e., consultants or experts in I&C software design and development for nuclear power plants) to support the software quality assurance activities.

[[

]]

3.3.1 NEW PLANT PROJECT (NPP) QUALITY MANAGER

The Quality Manager has the overall responsibility and authority of the GEEN QA program. The Quality Manager is also responsible for assuring that adequate resources are available to support the QA program, quality initiatives for improvement of processes used by GEEN for product and service offerings including but not limited to, reporting to top management on the performance of the quality assurance system and ensuring the promotion of awareness of quality requirements throughout the organization.

3.3.2 SOFTWARE QUALITY ASSURANCE MANAGER

The SQA Manager, who interfaces with the SPE Manager, has the overall responsibility and authority of the SQA Program. The SQA Manager is responsible for:

1. Approving this SQAP.
2. Approving or rejecting the validated software.
3. Issuing stop work order if the audit or assessment findings indicate violation of the quality and/or safety requirements.
4. Organizing the software auditing activities and maintaining the software audit plan.
5. Participating in baseline reviews.
6. Scheduling and coordinating software audits (both internally and externally) with the NPP Quality Team and/or the Nuclear Quality Assurance Team to ensure effectiveness of the audit being conducted.

Reporting audit results to the responsible project leadership (i.e., SPE Manager Engineering Manager, Project Management Team) and the Quality Manager.

3.3.3 NPP QUALITY MANAGER

The NPP Quality Manager has the overall responsibility and authority of the Quality Program for the ESBWR Project. The NPP Quality Manager shall coordinate with the SQA Manager concerning the audit of the software products. The NPP Quality Manager is responsible for:

1. Quality assurance requirements of the software products to assure completeness of quality control of the design and production of the software products. This includes but is not limited to:
   a. Hardware production.
   b. Hardware qualification.
   c. Shipping and packaging.
   d. Final product quality certification.
   e. Release for shipping approval.
2. Organization of the auditing activities and maintenance of the audit plan.

The NPP Quality Manager shall either reserve authority, or shall formally designate a Quality Control Engineer (QCE) who has the authority, to reject or order stop work when such action is deemed necessary to assure the quality or safety of the software product.

3.3.4 I&C AND ELECTRICAL SYSTEMS ENGINEERING MANAGER

The I&C and Electrical Systems Engineering organization, herein referred to as the Design Team, is described in the SMP [2.3(1)]. The I&C and Electrical Systems Engineering Manager has the overall responsibility to ensure the design and development of the software products is performed in accordance with the SMP [2.3(1)] and the required GEEN procedures and policies. This includes the approval or rejection of the design documentation, timely and effective control of work in process, and the quality of delivered software products.

3.3.5 SOFTWARE PROJECT ENGINEERING

The SPE is independent of the design team to ensure organizational freedom to perform the Quality tasks without undue pressure or conflict of interest related to budget and schedule. The following SPE teams are responsible for executing the Quality tasks described in this plan (section 3.2). A Task Lead is assigned by the SPE Manager to lead each of the following SPE teams:

1. Independent Verification and Validation Team (IVVT).
2. Software Safety Team (SST).
3. Baseline Review Team (BRT).

3.3.5.1 SPE MANAGER

The SPE Manager has the overall responsibility and authority for the implementation of the Quality tasks during the software life cycle. The SPE Manager is responsible for:

1. Coordinating with the SQA Manager in organizing the Quality tasks.
2. Approving the IV&V, SSA, or baseline review design outputs and documentation.
3. Rejecting the design outputs and documentation, as recommended by the Task Lead(s), if serious defects are identified during SSA and/or IV&V, such as, the requirements and/or design are incomplete, inconsistent, and not traceable to upper level documents.
4. Staffing of BRT, SST, and IVVT.
5. Appointing the Task Lead to BRT, SST, and IVVT.
6. Communicating open issues to Engineering organizations and the Project Management Team, and Quality Manager.
7. The overall management, including schedule and budget of SPE, to ensure continued effectiveness and support of the Quality tasks described in this SQAP.

3.3.5.2 INDEPENDENT VERIFICATION & VALIDATION TEAM

The IVVT is responsible for performing and managing Independent V&V (IV&V) tasks on the Software Class Q design documentation and design outputs to:

- Ensure that the design meets the specified requirements
- Confirm the quality, safety, reliability, availability, maintainability, testability, security, and performance of the design
- Ensure that the software products meet their intended use and do not perform unintended functions.

The IVVT Task Lead is responsible for:

- Organizing the IV&V tasks and coordinating the IV&V schedule with the Design Team
- Assigning IV&V tasks to IVVT Team members
- Managing the conduct of IV&V tasks, and reviewing and approving the IV&V reports prepared by IVVT Team members
- Ensuring that the IV&V is performed in accordance with the SVVP described in Section 7.0
- The Design Team is responsible for the independent verification of software class N3 and N2 software, as described in Subsection 3.3.1.1.

3.3.5.3 BASELINE REVIEW TEAM

The BRT is responsible for performing the baseline review to assess the adequacy of the software design process and control of configuration items (CIs) in accordance with the SCMP (Section 10.0). The BRT shall issue a Baseline Review Record for each baseline review conducted.

The BRT Task Lead is responsible for:

- Coordinating and scheduling the baseline review meetings with the Design Team and the SQA Manager
- Organizing the baseline review process
- Assigning tasks to BRT members
- Managing the conduct of BRT tasks
- Ensuring that baseline review tasks and activities are performed in accordance with the SCMP described in Section 10.0
- Approving the baseline configuration items
- Coordinating the release of configuration items into the Configuration Management System (CMS)

3.3.5.4 SOFTWARE SAFETY TEAM

The SST is responsible for performing the SSA to assure the safety characteristics of the software products being developed, including the interface between the hardware and software. The SST has the authority to enforce safety requirements in the software requirements specification (SRS), the design, and the implementation of the software.

The SST is responsible for determining the Software Class described in Section 1.5, Software Classification of the software and performing SSA as appropriate. The SST shall coordinate with the IVVT to evaluate the verification efforts to determine if SSA can be used as a verification method.

The SST Task Lead is responsible for:

1. Overseeing the overall conduct of the software safety program.
2. Organizing the software safety program and coordinating the SSA schedule with the IVVT and the Design Team.
3. Approving or rejecting safety software.
4. Assigning tasks to SST Team members.
5. Managing the conduct of SST tasks and approving SSA reports prepared by SST members.
6. Ensuring that each SSA is performed in accordance with the SSP described in Section 9.0.
7. Establishing supplemental tasks to support SSA.

3.3.6 CONFIGURATION MANAGEMENT MANAGER

The Configuration Management Manager (CMM) has the overall responsibility and authority for the CMS. Configuration Management is responsible for defining the CM process and tools, as well as execution of the CMS to maintain and control traceable records of:

1. Design Requirements and Inputs.
2. Design activities.
3. Design Output.
4. Authorizations to execute change requests to the controlled records.
5. Approvals of the execution of change requests.

3.3.7 DESIGN TEAM

The Design Team is responsible for the design and implementation of the software products and the independent verification and validation of Software Class N3 and N2 software products. The responsible verifiers and testers shall be individual(s) or groups(s) who are competent to perform verification and validation based on knowledge and experience. The verification and validation shall be conducted by individuals(s) or groups(s) other than those who performed the design of the software product. [[ ]] The roles and responsibilities of the Design Team are described in the SMP [2.3(1)].

3.4 Organizational Interfaces

Figure 2, SPE and Quality Organizational Functions Interfaces, depicts the interfaces between the SPE and the I&C and Electrical Systems Engineering, New Plant Project (NPP), Quality, and vendors.

The NPP Project Managers (PMs) are responsible for the commercial aspects of the software project, including:

1. Management and control of vendors' participation in the design and delivery of the software products.
2. Maintaining coordination between SPE and vendor organizations.
3. Vendors' performance have the authority to request audit to be conducted on the vendor who continues to violate quality requirements specified in the purchase order or contract.
4. Interface with customer (i.e. owner/user), herein referred to Licensee.

The detailed responsibility of the PM is described in the SMP [2.3(1)]. The SQA Manager, with support of the NPP Quality Team, shall perform SQA audits on the external vendor organizations prior to contract agreement (Section 14.1, Vendor Control). Vendors responsible for producing software products within the scope of the SQAP shall be in compliance with the requirements specified in this SQAP, including regulatory requirements described in the SMP [2.3(1)].

3.5 Scheduling and Planning

The SPE and SQA Managers have the overall responsibility for scheduling and planning the tasks and activities describe in this SQAP. The Task Lead for each team (SST, IVVT, BRT) is responsible for the management and planning activities for their respective teams. The Task Leads shall coordinate with the Design Teams concerning the timely receipt of design documentation to support the quality tasks (SSA, IV&V, baseline review, and software audit). [[

]]

As the Quality tasks are performed by a cross-functional team, a project workflow shall be established to ensure the required tasks are accurately identified and the Quality tasks schedule is aligned with the established integrated project schedule and milestones. The schedule shall:

1. Cover the duration of the SQAP.
2. Contain the major milestones of the project related to the Quality tasks.
3. Include the sequence and dependencies of the Quality tasks and the relationship of key Quality tasks to project milestones.
4. Express as absolute dates.

3.6 Approval Authority

The NPP Quality Manager, the SQA Manager, and the SPE Manager have approval or rejection authority for functions under his/her responsibilities.

Upon the rejection of a software product or the issuance of a stop work order, corrective actions shall be established, which may include a correction or amendment of the design process, revision to the software plans, re-design, re-implementation, or re-testing of the software

product. The Design Team shall be required to complete the corrective actions and identify preventive actions to avert the occurrence of similar defects.

4.0 DOCUMENTATION

The SMP [2.3(1)] establishes the managerial process and the technical direction necessary to govern the design and development activities of the software products. The required design documents and design outputs to be prepared are defined in the SMP [2.3(1)].

[[

]]

5.0 STANDARDS, PRACTICES, CONVENTIONS AND METRICS

5.1 Standards, Practices and, Conventions

The applicable Nuclear Energy EOPs and P&Ps used in guiding the design and development of software products are specified in Section 2.3, Supplemental Documents. If detailed instructions are needed, then project or platform/product line specific work practice instructions, such as ESBWR Project Instructions (EPI) or Engineering Service Instructions (ESI), are prepared to provide additional instructions as required. Software audits shall be conducted to monitor the compliance to the policies and procedures used to guide the design and development of software products.

Software coding shall be implemented in accordance with the guidelines defined in the Software Coding Conventions and Guidelines document required by the SMP [2.3(1)], which at a minimum, shall include

1. Documentation standards.
2. Logic structure standards.
3. Coding standards.
4. Commentary standards.

Code review shall validate the coding compliance to the guidelines outlined in the (applicable) Software Coding Conventions and Guidelines document.

5.2 Metrics

Software Metrics are sets of data which are systematically collected and analyzed in order to provide software quality process feedback to the software development processes. This feedback mechanism provides a means by which the software development processes can change over time to facilitate continuous process improvement with the primary objective of producing high quality defect free software products. Specific metrics will be defined for each software platform or product line and for each software classification.

The metrics program shall focus on the software functional and process characteristics listed in Appendix D. These characteristics will be used to derive a core set of metrics relating to the development process and the design documentation and outputs, such as requirements and design documents, code, and test documentation.

The SPE will be responsible for collecting and analyzing metric data for the software Class Q and N3 software products.

6.0 REVIEWS AND AUDITS

6.1 Technical Review

The purpose of the technical review is for a qualified individual, or a team of qualified individuals, to determine the suitability of the intended use of a design and identify discrepancies from design inputs, codes, and standards. It ensures the following:

1. The design document conforms to its specifications.
2. The design document adheres to regulations, standards, guidelines, plans, and procedures applicable to the project.
3. The design document is complete and correct.
4. For a document under revision, the changes have been implemented as specified in the change request or anomaly report.

Technical review may be conducted through peer review or design review.

Peer review shall be conducted by an individual other than the RE responsible for the design document. A Peer review is considered to be an informal review, thus cannot be used to replaced independent verification. The review comments shall be documented and dispositioned by the RE. The review comments shall be filed in the project DRF.

[[

]]

6.2 Managerial Review

[[

]] The review team shall assess opportunities for improvement and the need for changes to the SQA program and quality objectives. [[

]]The review shall be documented in the Managerial Review Report, which shall include decisions and actions needed to assure continued effectiveness of the SQA program. Maintenance of the SQAP is described in Section 18.0, SQAP Maintenance.

6.3 Project Closeout Review

The responsible PM shall schedule a post-delivery closeout review to formally terminate the activities of a project, such as closing any Corrective Action requests (CARs) associated with the project and project Design Record File (DRF), setting up warranty administration and review, and conducting a Licensee closeout meeting to solicit feedback, which includes collecting lessons learned and metrics during the project. [[

]]

6.4 Audits

6.4.1 FUNCTIONAL AUDIT

The functional audits shall be conducted to assure that the requirements specified in the System Design Specification (SDS) and SRS have been met by checking the applicable Requirements Traceability Matrix (RTM). The functional audit shall be performed during baseline review by the BRT and shall be documented in the Baseline Review Record. The functional audit shall be performed for the Software Class Q software products and recommended for Software Class N3 and N2 software products.

6.4.2 PHYSICAL AUDIT

The physical audit shall be conducted to verify the appropriate CI item, which include Software Build Description, has accurately and completely described the "build" parameters of the software such that a duplicate version of the object code can be recreated. The physical audit shall be performed as part of Test Baseline Review by the BRT and shall be documented in the Test Baseline Review Record. The physical audit shall be performed for Software Class Q software products and recommended for Software Class N3 and N2 software products

6.4.3 IN-PROCESS AUDITS

This SQAP requires SQA audits to be performed on the design organizations (both internal and external) who are working on the ESBWR Project. The SQA audit shall be performed (by the SQA) to ensure compliance with the codes and standards specified in this SQAP. The SQA audit evaluates the adequacy and completeness of the required reviews and V&V activities. [[

]]

An audit report shall be prepared at the conclusion of each software audit. The audit report shall summarize the audit activities and results, observations, Conditions Adverse to Quality (CAQs), discrepancies, non-compliances to the required quality and engineering procedures, and recommended corrective actions. [[

]]

7.0 SOFTWARE VERIFICATION AND VALIDATION PLAN

This SVVP establishes the V&V tasks for the software designed and developed for software products. This SVVP satisfies the requirements of RG 1.168 [2.2.3(1)], except where specified in Appendix A. RG 1.168 endorses IEEE Std. 1012, "IEEE Standard for Verification and Validation Plans" [2.2.4(1)] and IEEE Std. 1028, "IEEE Standard for Software Reviews and Audits" [2.2.4(2)].

7.1 Purpose and Scope

7.1.1 PURPOSE

The purpose of this SVVP is to outline the specific V&V steps required during the software development process to ensure that:

1. The developed software meets its specified requirements, performs its intended functions correctly, and does not perform any unintended function.
2. The final software product meets the contract requirements, required industry and regulatory standards, and licensing commitments.
3. The final software product is correct, complete, accurate, and traceable to requirements specified in the design documents and outputs.

The goal of this SVVP is to assure that software V&V activities are integrated throughout the software life cycle to facilitate the timely detection of errors and to ensure the quality of the software product.

7.1.2 SCOPE

This SVVP outlines the formal set of standards and procedures necessary to comprehensively verify and validate Class Q, and Class N3 and N2 software products during all phases of the software life cycle. The software life cycle phases in the SVVP correspond with those defined in the SMP [2.3(1)].

The V&V tasks, which cover by this scope, are described in Section 7.3. Software V&V activities shall also be included to analyze and test the software with respect to its hardware interfaces and user interactions. V&V activity is limited to software prepared by GEEN and GEEN vendors for the ESBWR project and evaluation of the qualification results of Commercial off-the Shelf (COTS) software to be used in a Software Class Q software product. Qualification of COTS software is performed by the Design Team as described in Section 5.7.6 of the SMP [2.3(1)].

7.2 Verification and Validation Overview

7.2.1 ORGANIZATION

Section 3, Management Organization describes the organization efforts in supporting the V&V activities.

7.2.2 V&V SCHEDULE

The V&V schedule and contingency planning to identify risks shall be documented in the IV&V Tasks PWP (Software Class Q) and the project PWP (Software Class N3 and N2).

7.2.3 SOFTWARE INTEGRITY LEVEL SCHEME

This SQAP uses an approach similar to the software integrity level scheme specified in IEEE 1012 [2.2.4(1)] to define the V&V requirements for the software product. IEEE 1012 does not mandate the use of the software integrity scheme specified in the standard. The approach used by this SQAP is described in Section 1.5, Software Classification.

7.2.4 RESOURCES SUMMARY

Subsection 3.3.5.2, Independent Verification and Validation Team, describes the personnel required to support IV&V activities for Software Class Q software. The SPE Manager is responsible for IVVT staffing and budget. The Design Team is responsible for the Software Class N3 and N2 software V&V.

Subsection 3.3.5.3, Baseline Review Team, describes the personnel required to support the baseline review activities.

Subsection 7.2.6, Tools, Techniques, and Methods, addresses the tools, techniques, and methods used to support the V&V activities.

7.2.5 ROLES AND RESPONSIBILITIES

The roles and responsibilities of IVVT members are described in Subsection 3.3.5.2, Independent Verification and Validation Team and Subsection 3.3.2, SQA Manager describes Quality Organization support in the V&V activities.

The Design Team is responsible for the Software Class N3 and N2 software V&V. The roles and responsibilities of the Design Team are described in the SMP [2.3(1)]. For Software Class Q software, the Responsible Technical Project Engineer (RTPE) shall formally notify the IVVT Task Lead via a formal project letter when a design document is ready for IV&V.

The RTPE shall formally notify the BRT Task Lead via a formal project letter when a software life cycle phase is ready for baseline review.

The project letters shall be filed in the project DRF.

Table 2 lists the V&V tasks and the individual or group responsible for performing these tasks.

7.2.6 TOOLS, TECHNIQUES, AND METHODS

7.2.6.1 V&V TOOLS

Tools used to support the V&V tasks shall be evaluated. The evaluation results shall be documented in the tool evaluation report. [[

]]

7.2.6.2 TECHNIQUES AND METHODS

7.2.6.2.1 VERIFICATION

Verification is performed to determine whether or not the design document/output for a given software life cycle phase fulfilled (i.e., is traceable) the requirements. [[

]]

7.2.6.2.2 CODE REVIEW

Code reviews are performed to verify that the software correctly implements the specified design [[

]]

7.2.6.2.3 SOFTWARE FUNCTIONAL TEST

The software functional test includes the software module/unit test and the software integration test. [[

]]

7.2.6.2.4 SOFTWARE VALIDATION TEST

The software validation test is performed to validate that the software product is operational and conforms to the functional and performance requirements [[

]]

7.2.6.2.5 BASELINE REVIEWS

Baseline Reviews are formal, independent evaluations of the software design and development activities performed at the completion of each software life cycle phase. [[

]]

7.2.6.2.6 REQUIREMENTS TRACEABILITY ANALYSIS

Requirements Traceability analysis (RTA) is performed for Software Class Q, N3 and N2 software requirements. [[

]]

7.2.6.2.7 AUDIT SUPPORT

Subsection 6.4.3 describes the in-process audit. [[

]]

7.2.6.2.8 WALK-THROUGH

Design walk-through is [[ ]] used during the design and development of the software product [[

]]

7.3 Verification and Validation Activities and Tasks

The following sections describe the V&V activities and tasks to be performed for each life cycle phase. [[

]]

9.0 SOFTWARE SAFETY PLAN

9.1 Purpose and Scope

This SSP establishes the processes and activities intended to ensure that the safety concerns of the software products are properly considered during the software development and are consistent with the defined system safety analyses as defined by RG 1.173, Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants [2.2.3(6)]. This SSP meets the guidelines specified in Chapter 7 of NUREG 0800 SRP [2.2.1(1)] 1 and the requirements outlined in section 4.4 of IEEE Std. 1228, "IEEE Standard for Software Safety Plans" [2.4(1)].

Safety is the most important consideration, taking precedence over budget and schedule.

SSA is performed on the software for Software Class Q and N3 and software products.

9.2 Software Safety Management

9.2.1 ORGANIZATION AND RESPONSIBILITIES

Section 3.0 describes the organization efforts in supporting the SSA activities. The roles and responsibilities of SST are described in Subsection 3.3.5.4.

9.2.2 DOCUMENTATION REQUIREMENTS

The documents to be prepared in support of the development of the SSA are specified in the SMP [2.3(1)]. A summary of the documentation to be prepared is listed below:

1. Software Project Management. Documentation of how the software safety program will be implemented, integrated, and managed with the software development activities is discussed in this SQAP and in the SMP [2.3(1)].
2. Software Configuration Management. Information regarding the CM of the design documentation and design outputs is specified in Section 10.0, Software Configuration Management Plan.
3. Software Quality Assurance. Information regarding the SQA of software design documentation and design outputs is specified in this SQAP.
4. Software Safety Requirements. Specification of safety requirements to be met by the software to avoid or control system hazards is specified in the SMP [2.3(1)] Subsection 5.7.8, Software Requirements Specification.
5. Software Safety Design. Descriptions of the software design elements that satisfy the software safety requirements are specified in the SMP [2.31)] Subsection 5.8.3.1, Software Design Description.
6. Software development methodology, standards, practices, metrics, and conventions. Approved and controlled practices that are essential to satisfy system and software safety objectives and requirements are specified in the SMP [2.3(1)] Subsection 5.8.3.4, Software Coding Conventions and Guidelines Document.

7. Test Documentation. Software safety-related test planning, test design, test cases, test procedures, and test reports are specified in Section 8.5, Test Documentation.
8. Software Verification and Validation. Information regarding how software safety will be verified and validated is defined in Section 7.0, Software V&V Plan. The software safety-related analyses are specified in this SSP. RTA (Subsection 7.2.6.2.6) is used to ensure the traceability of safety requirements to the design specifications, software source code, and software safety-related test cases.
9. Reporting Safety Verification and Validation. Information documenting the results of software safety-related verification and validation activities is specified in Section 7.4, Verification and Validation Reporting.
10. Software User Documentation. Information that may be significant to the safe installation, use, maintenance, and/or retirement of the software product is specified in the SMP [2.3(1)] Subsections User's Manual and 8.0, Software Operations and Maintenance Manuals.
11. Results of Software Safety Requirements Analysis. The reporting requirements for this activity are specified in Subsection 9.3.2, Software Safety Requirements Analysis.
12. Results of Software Safety Design Analysis. The reporting requirements for this activity are specified in Subsection 9.3.3, Software Safety Design Analysis.
13. Results of Software Safety Code Evaluation. The reporting requirements for this activity, including software functional testing, are specified in Subsection 9.3.4, Software Safety Code Analysis.
14. Results of Software Safety Test Analysis. The reporting requirements for this activity are specified in Subsection 9.3.5, Software Safety Test Analysis.
15. Results of Software Safety Change Analysis. The reporting requirements for this activity are specified in Subsection 9.3.7, Software Safety Change Analysis.

9.2.3 TOOL SUPPORT AND APPROVAL

Software tools used in the development and evaluation of software class Q and N3 software shall be evaluated for suitability. Software tools used to aid the development or evaluation of the software are managed as specified in the SMP [2.3(1)]. Configuration control of software tools is managed in accordance with the requirements of the SCMP (Section 10.0).

9.2.4 PREVIOUSLY DEVELOPED OR PURCHASED SOFTWARE

The SST has the authority to reject the use of the PDS and COTS software in Software Class Q and Quality Class N3 software if the PDS and COTS software does not meet the requirements of this plan.

9.2.5 PROCESS CERTIFICATION

Process certification is achieved through baseline reviews. Baseline reviews are described in Subsection 7.2.6.2.5.

9.3 Software Safety Analyses

The SSA is performed to ensure that:

1. The system safety requirements have been correctly addressed.
2. No unmitigated hazards have been introduced.
3. Software elements that can affect safety are identified.
4. There is evidence that other software and system elements do not affect safety.
5. Safety problems and resolutions identified in these analyses are documented.

The SSA shall be conducted to determine the Software Class as described in Section 1.5, Software Class. SSA shall focus on the software identified to have Software Class Q and N3.

[[

]]

9.4 Post development

9.4.1 TRAINING

The Software Training Plan is described in Section 9.0 of the SMP [2.3(1)].

9.4.2 DEPLOYMENT

9.4.2.1 INSTALLATION

Installation is described in Section 7.0 of the SMP [2.3(1)]. Installation V&V tasks are described in Subsection 7.3.7 and the SAT is described in Section 8.4.

9.4.2.2 STARTUP AND TRANSITION

Prior to starting up the newly installed software product, pre-operational tests shall be conducted to demonstrate the installed software product operates as intended, and if applicable, the required setpoints (e.g., trip and alarm) are established. The pre-operational test shall be conducted in accordance with an approved (by the Licensee) test plan and procedure. The pre-operational test is outside the scope of this SQAP as it is usually the Licensee's responsibility and shall be supported by qualified engineers who are knowledgeable in the installed software product and plant operation.

The Startup Procedure shall address the requirements for safely starting the new system and, if an old system is to be replaced, for making a safe transition from the old system to the new system. At a minimum, the following shall be addressed:

1. Fallback modes for the new system.
2. Startup of backup components and subsystems.
3. Startup of the new system.
4. Parallel operation with backups.
5. Parallel operation of the old system and the new system.
6. Subsystem vs. full system operation.
7. Switchover to full system operation.
8. Validation of results from the new system.
9. Cross validation of results between the old system and the new system.

10. Fallback in the case of failure of the new system, including fallback to an old system if one exists.

9.4.2.3 OPERATIONS SUPPORT

The Software O&M Manual and User Interface Specification shall be provided for the safety-critical software. The Software O&M Manual and User Interface Specification are described in Section 8.0 and Subsection 5.7.11 of the SMP [2.3(1)], respectively.

9.4.3 MONITORING

The Licensee is responsible for monitoring the operation of the safety-critical software within the software product. Safety concerns that are detected during operation shall be documented and reported in accordance with the plant's problem reporting procedures. [[

]]

9.4.4 MAINTENANCE

Software maintenance is specified in the Software O&M Manual. The software O&M Manual is described in Subsection 5.11.8 of the SMP [2.3(1)].

9.4.5 RETIREMENT AND NOTIFICATION

Retirement and notification are described in Subsection 5.13 of the SMP [2.3(1)].

9.4.6 PLAN APPROVAL

[[ ]]
The review basis for this plan is the ESBWR MMIS/HFE IP [2.1(1)] and RG 1.173 [2.2.3(6)]. The SPE manager shall have approval authority to approve this plan upon completion of the design review.

9.5 Software Safety Analysis Report

As a minimum, the software safety analysis shall include the following:

1. Name and description of the software evaluated.
2. System.
3. Software Classification.
4. Purpose and scope.
5. Reference inputs.
6. Software Safety Analysis body of report.
7. Anomalies noted.
8. Conclusion.
9. Responsible Engineer.
10. Approving Authority.

The report shall be placed under the configuration control as described in Section 10.0, Software Configuration Management Plan.

10.0 SOFTWARE CONFIGURATION MANAGEMENT PLAN

10.1 Introduction

This SCMP establishes the SCM activities for the design and development of the software products. This SCMP satisfies the requirements of RG 1.169, Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants [2.2.3(2)], except where specified in Appendix A. RG 1.169 endorses IEEE Std. 828, IEEE Standard for SCM Plans [2.2.4(3)].

10.1.1 PURPOSE

The intent of this SCMP is to provide additional guidance and direction necessary to implement the SCM activities required during the software product design and development process. This SCMP supplements GEEN established configuration management procedures in system and hardware design. It establishes a formal set of standards and methodology used to administer and control the configurations of Software Class Q and Software Class N3 and N2 software products and shall remain in effect throughout the software life cycle phase.

10.1.2 SCOPE

The scope of SCMP includes the following:

1. SCM Management. SCM Management describes the individual with the overall responsibility and authority for the SCM and organizations responsible for supporting the SCM activities.
2. SCM activities. SCM activities define the SCM tasks, including methods, timing, and responsibility for the implementation of design control and design change control.
3. SCM schedule. SCM schedule identifies the SCM required schedule and coordination with the design activities and the Quality tasks described in this SQAP.
4. SCM Resources. SCM resources identify the tools, procedures, and individuals needed to execute or support each SCM task.

10.2 SCM Management

10.2.1 ORGANIZATION

The hierarchy of responsibility for the SCM activities is as follows:

1. Configuration Management Manager. The Configuration Management Manager (CMM) has the overall responsibility and authority for the CMS, including system maintenance and enhancement.
2. BRT Task Lead. The BRT Task Lead has the overall responsibility of the baseline review process and the configuration control of software products.
3. Responsible Configuration Control Engineer (RCCE). The configuration control engineer is responsible for the configuration control of the design documentation and outputs related to the software product, and the maintenance of software library.

4. Baseline Review Team. The BRT is responsible for judging adherence to the software development process for the design documentation and/or outputs being baselined. The members of this team are appointed by the BRT Task Lead and must be independent from the design team responsible for the design documentation and/or outputs.
5. Responsible Manager. The responsible manager is responsible for the technical scope (design and development) of the software product.
6. Responsible Engineer. The responsible engineer is responsible for a given technical item (e.g., the design and development of the documentation).

10.2.2 SCM RESPONSIBILITIES

1. The primary responsibilities of the Configuration Management Team, under the direction of the CMM, supports the
   a. Design control throughout the software life cycle phase to ensure compliance with the applicable safety and performance requirements.
   b. Design change control to establish the change approval criteria for high impact change requests [[ ]]
   c. Engineering document management to control the project records.
   d. Engineering document format and issuance to ensure consistency and standardization of the engineering documentation and issuance process are being used and followed.
2. The Change Control Board (CCB) is responsible for the evaluation of the proposed high impact modifications to the software product design or product configuration documentation and provides recommendations, which include concurrence, rejection, modification, or hold for further investigation. [[ ]]

   High impact modification is a change that affects one or more of the following factors:
   a. Safety and licensing.
   b. System or plant performance.
   c. Design interface (internal or external). [[ ]]
3. The SCM responsibilities of the I&C and Electrical Systems Engineering Manager is to review and approve the initiation or change of design documents and Software Class N3 and N2 software to confirm:
   a. Review and verification were performed by technically competent individual(s).
   b. Scope of review and verification [[ ]]
   c. Comments made by the reviewers were adequately resolved.
4. The SCM responsibility of the responsible SQA manager:
   a. Approve/reject the validated software.
   b. Participate in Baseline Reviews.

5. The SCM responsibilities of the TPE are to:
   a. Identify the reason for the document initiation or change (i.e., error correction, regulatory or Licensee requirement, etc.).
   b. Determine the timing of baseline review.
   c. Identify the items to be baselined.
   d. Authorize the distribution or release of baselined (validated) software (i.e., source, object, and executable codes) to RCCE for configuration control.
6. The SCM responsibilities of the RE are to:
   a. Initiate or revise engineering controlled documentation and obtain the required verification documents [[
      ]]
   b. Resolve the non-conformances identified.
7. The responsibilities of the RCCE are to:
   a. Ensure the software and associated documentation are entered into the software library after the approval of the BRT.
   b. Release software source code/application code to the Design Team for revision or the approved software package for production.
   c. Coordinate software configuration control with Configuration Manager.
   d. Support baseline review as a BRT member.
   e. Maintain the software library.
8. The BRT chairperson is appointed by the BRT Task Lead. The responsibilities of the BRT chairperson are to:
   a. Appoint members of the BRT.
   b. Establish the BRT by assigning review responsibilities.
   c. Convene baseline reviews.
   d. Chair the baseline reviews.
   e. Document the baseline review meeting, BRT members, and attendees. This information shall be stored in the appropriate DRF.
   f. Track open baseline items.

The BRT members shall have sufficient skill and experience to effectively judge the adequacy of the V&V of the CIs being baselined. The BRT Members shall be knowledgeable in the Baseline Review process. They shall be independent from the design and development of the CIs under review.

The BRT may be comprised of individuals from the Quality Organization, the Configuration Management or they may be members of the design team that are independent of the design and development of the CI subject to baseline. The responsibilities of the BRT are to:

a. Ensure that the CIs are properly identified, verified, and controlled.

b. Ensure compliance with the SMP, SCMP, SVVP, and SSP.

c. Review and approve the resolved nonconformance comments from baseline reviews.

The project manager is responsible for maintaining the coordination of releasing issued design documentation to the vendor supporting the project, including the coordination of review and approval by the RE of vendor submittals, the design interface review, and control of project correspondence. Vendor control is described in Section 14.1.

10.2.3 APPLICABLE POLICIES, PROCEDURES, AND DIRECTIVES

The Nuclear Energy P&Ps, EOPs, and directives applicable to the SCM activities, are specified in Section 2 and its subsections. These policies and procedures are used to supplement the process specified in this SCMP. If any external constraints are placed on the plan per contract requirements, such constraints and its impact and effect on the SCMP shall be documented in the project PWP.

10.2.4 SCM SCHEDULE

The SCM schedule which establishes the sequence and the identified SCM tasks shall be specified in the PWP of the individuals responsible (Subsection 10.2.1) for the SCM tasks. Section 3.5 describes the scheduling and planning for the Quality tasks, including the tasks related to SCM.

10.3 SCM Resources

10.3.1 SCM TOOLS

The following are the SCM Tools used to support the design of software products:

1. Product Data Management System (PDMS) is the GEEN official CMS. It is used for the creation, control, approval, storage, and retrieval of documents or data in electronic media. PDMS is described in Subsection 12.1.4, Product Data Management System
2. Design Record File (DRF) is a formal controlled information record under the GEEN procedures for in-progress and completed engineering work which is retained and from which work can be retrieved. DRF is described in Subsection 12.1.5, Design Record File
3. HFE Issue Tracking System (HFEITS) is an web based database used to track:
   a. HFE issues that are not resolved through the normal HFE process.
   b. Software problems, defects, or anomalies discovered during design and development (not part of V&V activities).
   c. Baseline review open items.
   d. HFEITS is described in MMIS/HFE IP [2.1(1)].

4. Commitment Tracking System (CTS) (Subsection 12.1.1) is used to track:
   a. Requirements, deviations, repeat procedural violations, and non-conformances.
   b. Post delivery software and documentation errors and discrepancies.
   c. Issues identified that are outside the scope of the Design Review, [[
      ]]
5. Discrepancy Tracking System (Subsection 12.1.6) is used to track anomalies identified during software validation test, [[ ]] and applicable, SAT.

10.3.2 SCM TECHNIQUES

Section 10.4 identifies the SCM Tasks, the techniques and procedures used to accomplish each task, the individuals responsible for each task, and applicable tools used to support each task.

10.4 SCM Tasks

10.4.1 CONFIGURATION IDENTIFICATION

The CIs subject to this plan include:

1. Engineering documents prepared to document the design, to communicate system requirements for software products and material, and to support the implementation and manufacturing of the software products. Engineering documents are typically issued documents. They shall be assigned a unique document identification number, revision status, quality classification, and pagination, including the total number of pages in the document. [[

   ]]
2. Quality records such as V&V reports (Section 7.4), test reports (Subsection 8.5.3), audit reports (Subsection 6.4.3) and SSA reports (Criticality Analysis Reports and Hazard Analysis Reports described in Subsections 9.3.1 to 9.3.7) are prepared to document evidence of the quality of CI and/or execution of Quality tasks. These records shall be filed in the project specific Design Record File (DRF). A DRF is the formal controlled information record used to document design activities and retain/protect completed engineering work.[[

   ]]
3. Acquired software such as support software/tools, COTS software, and PDS, shall be assigned a unique identification and revision number in accordance with the format described in project level documentation.
4. Software, such as source code listings, objects, and executable files shall be assigned a unique file name and revision number.

   Each software module/unit source files shall contain a header comment section, which as a minimum, shall include the quality classification and a revision status. For Software Class Q software, reference design inputs documentation with a revision level shall also be included.

Upon completion of software validation testing, the validated software package, such as source code, object code, executable code, and associated data files for each released revision shall be placed under configuration control by the RCCE, or the individual assigned this responsibility, in the software library assigned for each specific product line or project. The software library shall serve as the final control point and repository for the released computer-based software configuration items. Different software libraries or project-specific software libraries may be used to control the computer-based configuration items as the software products may be implemented using other control product lines or other computer platforms. Procedures shall be established to describe the retrieval and reproduction process of the controlled computer-based software CIs from library storage.

The software library structures shall have a consistent naming convention, and an appropriate level of security control (e.g., password control). The security measures implemented shall provide assurance that the integrity of the baselined CI is maintained. Read and write control access to the software library accounts shall be granted to the RCCE. Personnel participating in the design and development of the software product shall only have read access to the software library. Changes to the software libraries can only be made by the Configuration Control Engineer.

Figure 5 presents an example the naming convention.

5. Vendor submittals shall be assigned a unique identification and revision number. [[

]]

All CIs shall be placed under configuration control and stored in the PDMS (Subsection 12.1.4). Table 4 presents a list of CIs, their structures, retention medium, and life cycle control points.

10.4.2 CONFIGURATION CONTROL

10.4.2.1 DESIGN CONTROL

The design of software products is controlled to ensure compliance with the applicable safety and performance requirements. Design control measures are established to achieve the following:

1. Definition of design requirements and performance of design activities in a planned, controlled, and orderly manner.
2. Specification of appropriate quality requirements and standards in design documents.
3. Selection of appropriate design verification methods and implementation by individuals or groups not directly responsible for the original design.

The design process performed shall be composed of the activities necessary for the complete software life cycle. The design process activities include analyses, preparation of specifications and drawings, testing, generation of test reports, and the technical support (i.e., installation and training) required to complete the design, implementation, installation, operation, and maintenance of the software products.

[[

]]

10.4.2.2 DESIGN CHANGE CONTROL

The design change control process includes change initiation, review, approval, implementation, disposition, status reporting, document updating, and distribution. The purpose of this process is to:

1. Ensure that total impact is considered before a change is approved.
2. Ensure that the documents are identified and changed after a change is approved.
3. Provide authority for a change.
4. Identify pertinent interfaces and organizations responsible for these interfaces.
5. Provide accurate and traceable records of change.
6. Ensure a schedule for implementation of approved design changes is established.

A change request may be initiated by the Licensee for product enhancement, or by anyone observing a problem/error with a software product, as described below. Reasons for a proposed change are categorized in Table 10.4.2.2-1.

Design change that results in modification to the ESBWR certified design shall be processed in accordance with section 3.1.4.2 of the MMIS/HFE IP [2.1(1)].

Table 10.4.2.2-1 Reasons for Change Request

Life Cycle Phase	Reasons
Requirements, Design, Implementation	1. Design requirements, 2. Change in regulatory requirement or codes and standards requirement
Tests	Anomaly or error correction during V&V and testing
Installation	Anomaly or error correction during installation
Operation and Maintenance	1. Anomaly or error correction during operation and maintenance 2. Licensee contract change
Requirements, Design, Implementation and Test	Change request from Vendor (Section 14.1 Vendor Control)

Table 10.4.2.2-2 Change Process Steps

[[

	]]

[[

]]

10.4.2.2.2 CHANGE REQUEST DURING V&V AND TEST PHASE

Subsection 7.5.1 describes the documentation of discrepancies or errors discovered during V&V and testing process. The discrepancies or errors shall be evaluated and resolved. If the discrepancies or errors impact an issued document or multiple documents, the RMCN process or the ECA process described above shall be followed.

10.4.2.2.3 CHANGE REQUEST DURING INSTALLATION PHASE

Discrepancies or errors discovered in a software product during the installation phase shall be processed using the Field Deviation Disposition Request (FDDR) process. [[
]]

10.4.2.2.4 CHANGE REQUEST DURING OPERATIONS AND MAINTENANCE PHASE

Change requests initiated during the Operations and Maintenance Phase as the result of software errors shall be reported and tracked CTS (Subsection 12.1.11).

10.4.2.2.5 CHANGE REQUEST FROM LICENSEE

Proposed changes to a software product design due to contract revision before product turnover or during the design and development process shall be processed using the ECA process. [[
]]

10.4.2.3 CHANGE NOTIFICATION

If the result of the discrepancies or errors affects products already installed or turned over to the Licensee, it shall be the responsibility of the Project Manager to:

1. Notify the affected plant Licensee of any detected non-conformances
2. Supply to the affected plant the upgraded software or EPROMs

10.4.2.4 DESIGN INTERFACES CONTROL

Engineering design interfaces with vendors or design organizations supporting the design of the software product shall be formally conducted and information formally transmitted. Project correspondence that pertains to the transmission or acceptance of project documents shall be maintained in PDMS.

To ensure interface compatibility, design documents shall be distributed for information and/or review to the affected design organizations to ensure that there is no conflict in the design objectives and to ensure that the product resulting from the interfacing designs function as planned. The PM is responsible for maintaining coordination of distribution of design documents to appropriate design organizations.

[[

]]

10.4.3 CONFIGURATION STATUS ACCOUNTING

Status for design documentation and design outputs can be collected from the PDMS by selecting report module feature to obtain the status of the design documentation and design outputs. The responsible TPE shall maintain a record or database used to prepare reports on the status of design documentation and design outputs. The record or database shall include initial approved version, the status of requested changes, and the implementation status of approved changes of each CI, as well as outstanding engineering documents undergoing engineering change requests that have not yet been resolved. Configuration status reports shall be used as supporting information to the project progress report to ensure timely reporting of project progress and baseline review.

10.4.4 CONFIGURATION AUDITS

Configuration audits shall be performed on the software CIs (including the computer-based items) to ensure the completeness of the software products. There are two types of configuration audits:

1. Functional Configuration Audit.
2. Physical Configuration Audit.

10.4.4.1 FUNCTIONAL CONFIGURATION AUDIT

A functional configuration audit is performed during the baseline review. The BRT shall inspect the design documentation, outputs, and associated traceability matrix for completeness (i.e., demonstration of forward and backward direction). Deficiencies shall be documented in the functional configuration audit minutes and maintained as an attachment or part of the Baseline Review Record. The responsible TPE is responsible for ensuring that the deficiencies are corrected.

10.4.4.2 PHYSICAL CONFIGURATION AUDIT

A physical configuration audit is performed during the Test phase baseline review. The BRT shall inspect the Software Build Description of the Software Class Q for completeness, such that a duplicate version of the software package can be recreated. The BRT shall also determine that all items identified as being part of the configuration are present in the product baseline. The audit must establish that the correct version and revision of each part are included in the product baseline and that they correspond to information contained in the baseline's configuration status report. Deficiencies shall be documented in the physical configuration audit minutes and maintained as an attachment or part of the Baseline Review Record. The responsible TPE is responsible to ensure that the deficiencies are corrected.

10.4.5 BASELINE REVIEWS

The baseline review is conducted at the completion of each software life cycle phase. For the O&M and Retirement phases, a review of baseline records is performed and revisions are made accordingly. This activity constitutes the baseline review for those phases. The following baselines have been designated by the SMP [2.3(1)]:

1. Planning.
2. Requirements.
3. Design.
4. Implementation.
5. Test.
6. Installation.
7. Operation and Maintenance.

The SMP [2.3(1)], in conjunction with the project PWP, specifies the CIs to be baselined during each software life cycle phase.

The purpose of the baseline review is to establish that:

1. The design information developed during the software life cycle phase adheres to the software life cycle process outlined in the SMP.
2. The V&V tasks and the SSA tasks performed adheres to the procedures outlined in the SVVP and SSP, respectively.

The baseline review is performed as follows:

1. Upon completion of the design activities within the software life cycle phase, including the required V&V tasks and SSA, the responsible TPE appoints an engineer to prepare the baseline package. The baseline package consists of CIs to be baselined for the specific software life cycle phase.
2. The responsible TPE shall notify the BRT Task Lead that the design activity of the specific software life cycle phase is completed and is ready for baseline review. The BRT Task Lead shall schedule the baseline review and convene a BRT.

3. The BRT shall be provided with the copies or the depository location of the CIs to be baselined (including the associated V&V reports) prior to the baseline review meeting.
4. A baseline review is performed to assess the design control and design change control, SSA, and V&V tasks of a particular software life-cycle phase.
5. The BRT has the authority to approve or reject the CIs to be baselined. The non-conformances and assessment shall be documented in the BRR (Subsection 7.4.5). The engineer responsible for the baseline package is responsible for resolving these non-conformances. The final resolution of the identified non-conformances shall be documented in the BRR.
6. A baseline review is not complete until the discrepancies have been resolved. However, if the responsible TPE can justify that the discrepancies discovered do not impact the safety and/or security requirements, exception may be granted at the discretion of the BRT to allow the design team to proceed to the next software life cycle phase. This justification must be documented in the BRR or as an attachment to the BRR.
7. The BRT task lead shall prepare the BRR. A copy of the BRR shall be forwarded to the responsible TPE to be filed in the software project DRF.

As software design and development is an iterative process, the baseline review shall be repeated as the baselined CI was modified.

10.4.5.1 BASELINE ITEMS APPROVAL PROCESS

The configuration items to be baselined must be reviewed by the BRT to confirm that:

1. Adherence to the SMP and SQAP has been achieved.
2. The required documents have been completed and verified.
3. The verification scope and approach is reasonable.
4. Any comments made during the review process have been adequately documented and that the non-conformances noted have been resolved.
5. The required testing has been completed, the results documented and verified, and the open issues have been resolved and approved by the BRT.

10.4.5.2 BASELINE REVIEW RECORD

The BRT chairperson shall prepare a BRR. Figure 4 provides an acceptable format for the Baseline Review. The BRR is described in Subsection 7.4.5.

10.5 Software Release Procedures

The RCCE has the responsibility and authority for the release of the approved (by SQA Manager) software package for production. The approved software shall be recreated in accordance with the procedures outlined in the Software Build Description.

10.6 Software Product Release

The responsible project QCE has the authority for the release of the final software product. The Software product is formally released for shipment upon issuance of Product Quality Certificate (PQC).

10.7 Vendor Control

Vendor control is described in Subsection 14.1.

10.7.1 SOFTWARE DEVELOPED BY VENDORS FOR THE PROJECT

The vendor shall utilize this SCMP to support the design and development of the software products or prepare an equivalent SCMP in accordance with the requirements outlined in this plan and the SMP. [[

]]

10.7.2 ACQUIRED SOFTWARE

Acquired software is maintained and controlled in accordance with the procedures outlined in Section 14.2.

10.7.2.1 CONFIGURATION CHANGE CONTROL OF ACQUIRED SOFTWARE

Acquired software may be modified by the supplier to:

1. Correct discrepancies or deficient conditions.
2. Improve performance.

If necessary, the RE shall reapply the evaluation process outlined in the SMP [2.3(1)] to the modified acquired software.

After the required evaluation has been performed and the revised evaluation report and test results have been verified in accordance with the methods outlined in the SVVP, the acquired software, with its associated documentation package, shall be:

1. Assigned a new revision number.
2. Baselined and placed under configuration control.

10.8 Record Collection and Retention

The baselined configuration items stored on a magnetic or optical medium shall undergo periodic archival backup [[
]] prescribes the requirements, procedures, and responsibilities for the control, retention, and retrieval of quality-related computer-based data maintained within the central computing facility of GEEN. All configuration items shall contain a direct indication of the item's revision status.

11.0 PROBLEM REPORTING AND CORRECTIVE ACTION

11.1 Problem Reporting

Discrepancies, deficiencies, anomalies, deviations or comments discovered during design and development (i.e. V&V, SSA and testing), installation, post delivery, and other Conditions Adverse to Quality (CAQs) shall be formally documented. Table 3 outlines the problem reporting process, including possible scenarios, responsible individuals, and documentation of reported problems. Defects and noncompliance Under 10 CFR Part 21 shall be reported. [[

]]

11.2 Corrective Action

It is essential that the process requirements described in this SQAP, the SMP [2.3(1)], the required EOPs, P&Ps, and Corporate QA program be adhered to. Failure to comply with these requirements shall be promptly identified and action shall be taken to eliminate or correct the nonconformities or CAQs to prevent recurrence. CAQs can be:

1. Discovered during Work performance and Audit.
2. A complaint from Licensees.
3. Findings from regulatory authorities.
4. Other external organizations (e.g., ISO/ASME Code authorities).

[[

]]

12.0 TOOLS, TECHNIQUES AND METHODOLOGIES

12.1 Tools

The SPE organization shall employ the use of tools as needed to execute the tasks specified in this plan. Tools used in part to perform V&V tasks do not need to be qualified if V&V is performed on the output produced by the tool.

12.1.1 COMMITMENT TRACKING SYSTEM

The Commitment Tracking System (CTS) is used to manage and record the identified Conditions Adverse to Quality (CAQs) and non-compliances to the established quality procedures, such as EOPs, P&Ps, and this SQAP as defined in Section 2. CTS is a web-based system residing on the GEEN intranet.

12.1.2 CHECKLIST

A checklist may be used to support inspection, independent verification and software audits to ensure completeness of the design output being verified or inspected, and the process being audited. The checklists prepared to support software inspection and independent verification should include acceptance criteria for the design output. The NUREG 0800, SRP [2.2.1(1)] divides the acceptance criteria into two sets:

1. Functional characteristics (accuracy, functionality, reliability, robustness, safety, security, or timing). Not all characteristics occur for every design output.
2. Process characteristics (completeness, consistency, correctness, style, traceability, un-ambiguity, or verifiability). Not all characteristics occur for every design output.

Software audits are conducted to independently evaluate the design team compliance with the SQA requirements specified in this SQAP and the applicable standards, regulations, guidelines, and procedures. The checklists prepared to support software audits should include queries to demonstrate compliance with the SQA requirements specified in this SQAP and the applicable standards, regulations, guidelines, and procedures.

12.1.3 REQUIREMENTS TRACEABILITY MATRIX

The Design team is responsible for the preparation of RTM. RTM can be prepared manually or using an automated tool. [[

]]

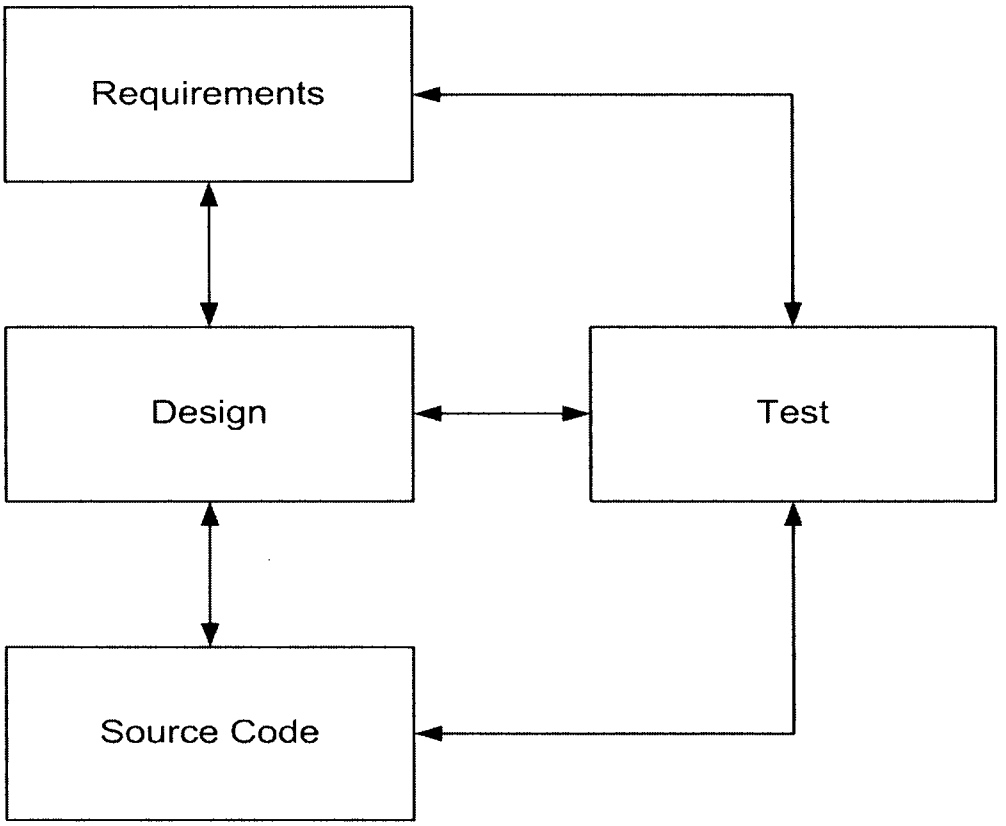


Figure 3. Example of a Traceability Matrix Structure

12.1.4 PRODUCT DATA MANAGEMENT SYSTEM (PDMS)

The Product Data Management System (PDMS) is an access controlled, computer-based data storage and retrieval system that is used to manage data relevant to the engineering definition of products and services, including quality records. It maintains previous and current revisions of the engineering documents that have been approved for issue. [[

]]

PDMS is the GEEN official CMS for engineering and quality controlled documents. Internal and external vendors providing the ESBWR software products are not required to utilize PDMS. However, an appropriate computer-based CMS shall be used.

12.1.5 DESIGN RECORD FILE (DRF)

A DRF is the GEEN formal controlled information record for in-progress and completed engineering work, which includes the design or modification of systems, hardware and software, and the performance or modification of analysis, evaluations, calculations, and licensing services, is retained and from which information can be retrieved. A DRF is an in-progress record that is contained in the PDMS and that is subject to change until it becomes a permanent Quality Assurance Record. [[

]]

12.1.6 DISCREPANCY TRACKING SYSTEM

A discrepancy tracking system will be initiated to manage and track the anomalies identified during software validation test, [[ ]] and SAT (if SAT is within GEEN scope of responsibility). Data pertinent to allow the RE to evaluate the anomaly and the responsible engineering manager to approve an anomaly are included in the discrepancy tracking system, which include:

1. Name and Master Parts List (MPL) of the software product.
2. Project / Plant.
3. Software classification.
4. Description of the anomaly, including effect and extent of the anomaly, clear explanation of observations, symptoms, workarounds, and any other pertinent information.
5. Severity of the anomaly.
6. Initiation date.
7. Affected documentation.
8. Affected design organization.
9. Corrective action / resolution statement.
10. Completion and approval status.

If needed, reports can be generated to:

1. Facilitate and monitor the anomaly disposition efforts
2. Ensure that the required changes to the affected design documentation and output has been completed.
3. Support baseline review and management review efforts.

12.2 Techniques and Methodologies

Techniques and methodologies used to support the Quality tasks are described in the SVVP (Section 7.0), SSP (Section 9.0), and SCMP (Section 10.0).

13.0 CODE AND MEDIA CONTROL

The computer-based design outputs, such as software source code, COTS software, and support software/tools used to support the design and development of software products are CIs and as such, shall be controlled as specified in the SCMP (Section 10.0).

14.0 VENDOR AND ACQUIRED SOFTWARE CONTROL

14.1 Vendor Control

Vendor selection and qualification shall be performed under a prescribed process. As a minimum, the following requirements shall be evaluated:

1. Ability to meet engineering, quality, and purchasing requirements.
2. Relevant experience in the design and development of similar products.
3. Awareness of and compliance with the applicable regulatory and industrial requirements.
4. Service, installation and support capability and history of performance.

Confirmation of this ability is determined by audits and/or reviews of the vendor's Quality Management System, including the Quality Assurance Program. The IVVT shall support the SQA during vendor audit.

[[

]]

14.2 Commercial Off The Shelf (COTS) Software

COTS software is software commercially available to anyone. COTS software includes communication protocol applications and linkable software libraries. It is acceptable that the qualified and dedicated COTS software be used in the Software Class Q software products. The SMP [2.3(1)] describes the qualification and dedication of COTS software.

14.3 Previously Developed Software

PDS is software developed for prior projects and not necessarily verified and validated per the requirements outlined in this SVVP. The IVVT shall independently verify the PDS evaluation report prepared by the design team for software intended to be used in Software Class Q software product. The SMP [2.3(1)] describes the evaluation and dedication of PDS.

15.0 RECORDS COLLECTION, MAINTENANCE, AND RETENTION

Section 10.8, Record Collection and Retention describes the collection, maintenance and retention of design documentation, design outputs and quality records, such as audit reports, SSA reports, and test reports.

16.0 TRAINING

All personnel supporting the Quality tasks shall be trained, as necessary, to ensure proficiency in applicable quality and technical tasks prior to the assignment of work activities affecting the quality of software products [[

]] The design team and the SPE teams shall be trained, either by self-study or classroom, in this SQAP, the SMP [2.3(1)], applicable tools required to support the design and V&V tasks, and the referenced EOPs and P&Ps. The training records shall be maintained in appropriate ESBWR and GEEN training databases.

17.0 RISK MANAGEMENT

Risk Management is the process of identifying, controlling, and eliminating or minimizing uncertain events that may affect the project. [[

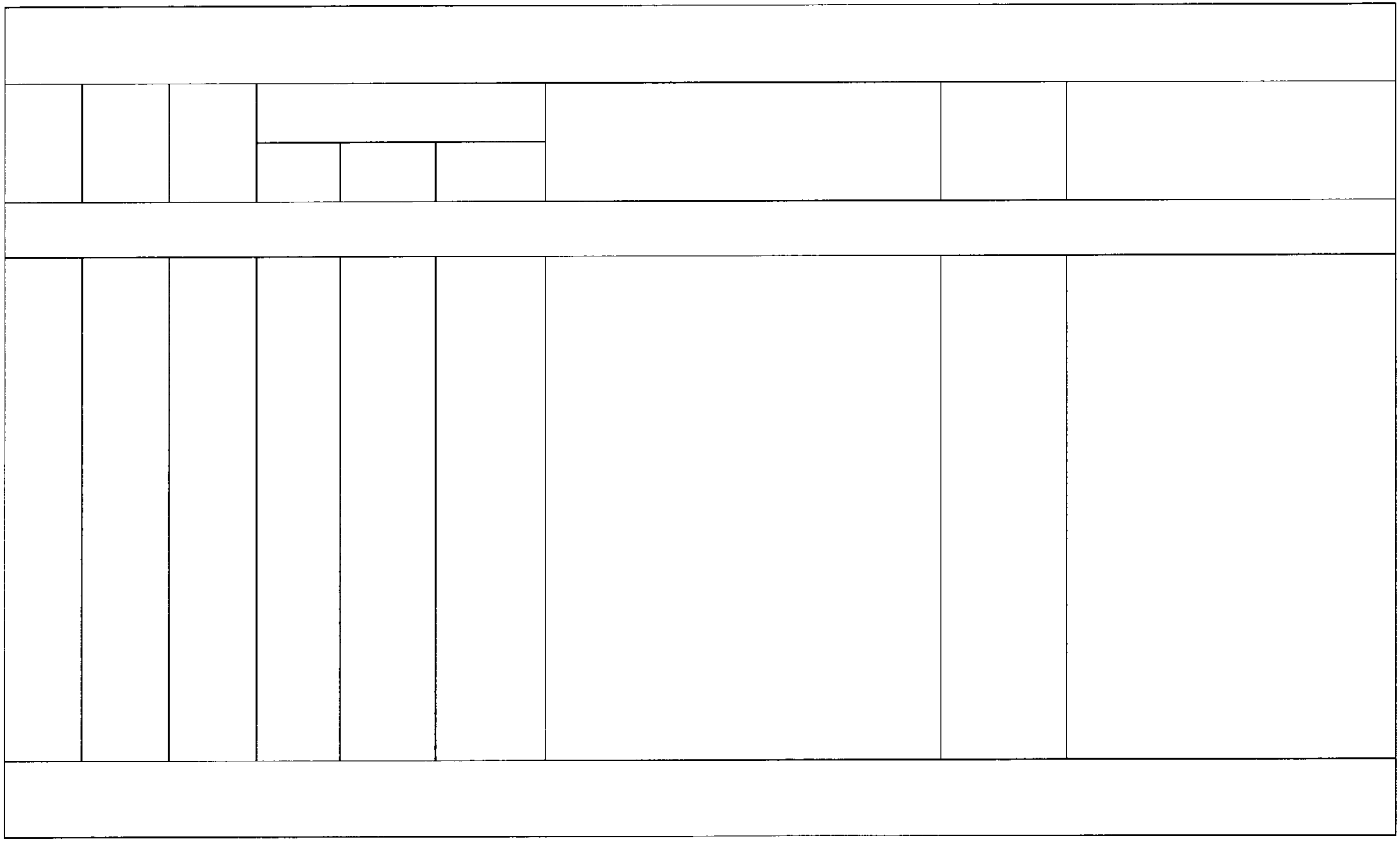
]] The Task Leads shall prepare a risk management plan to document responsibilities and actions needed to assess, abate, monitor, and control the identified risks and concerns. It is acceptable that the risk management plan be included in the task specific PWP.

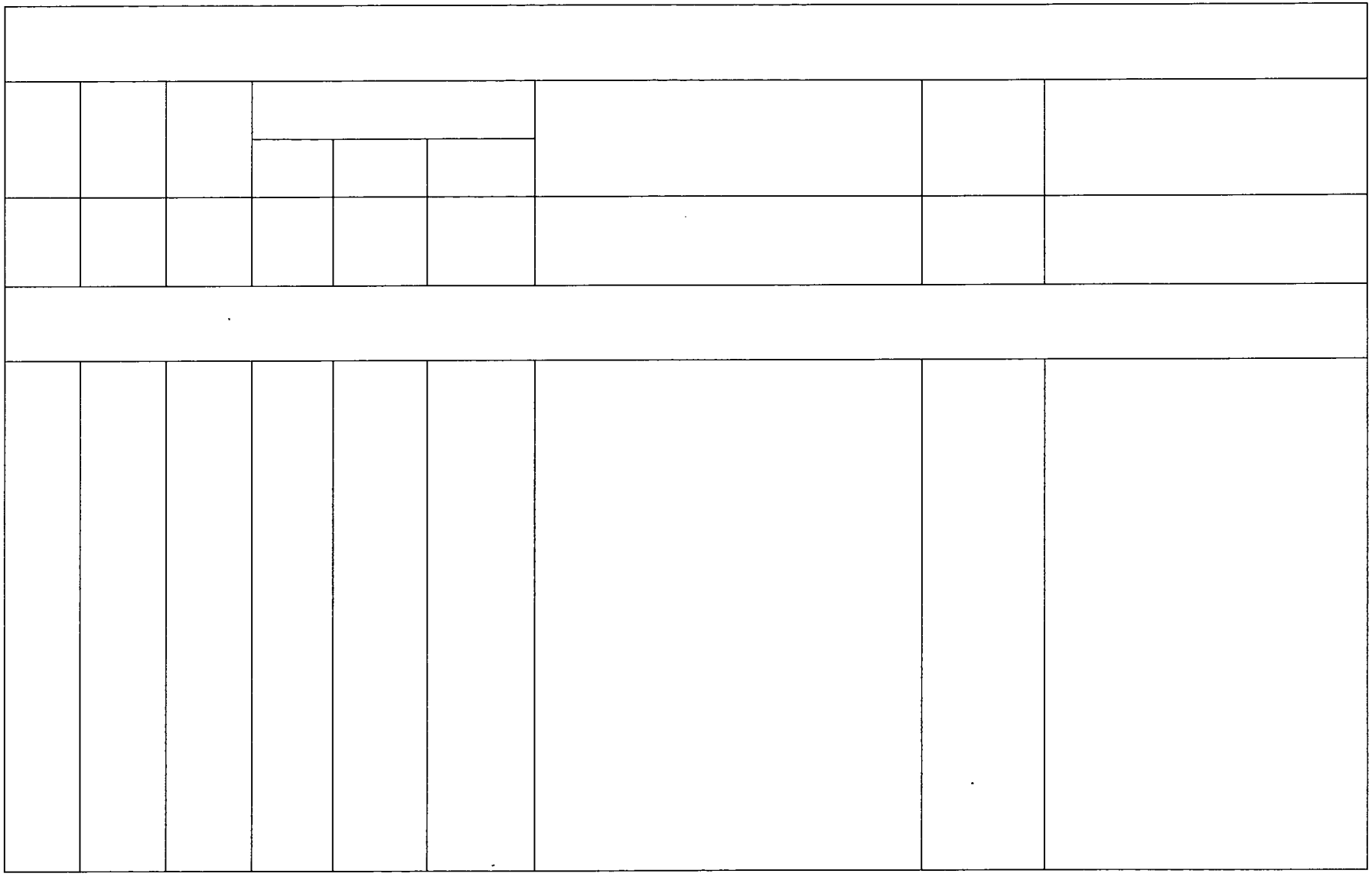
18.0 SQAP MAINTENANCE

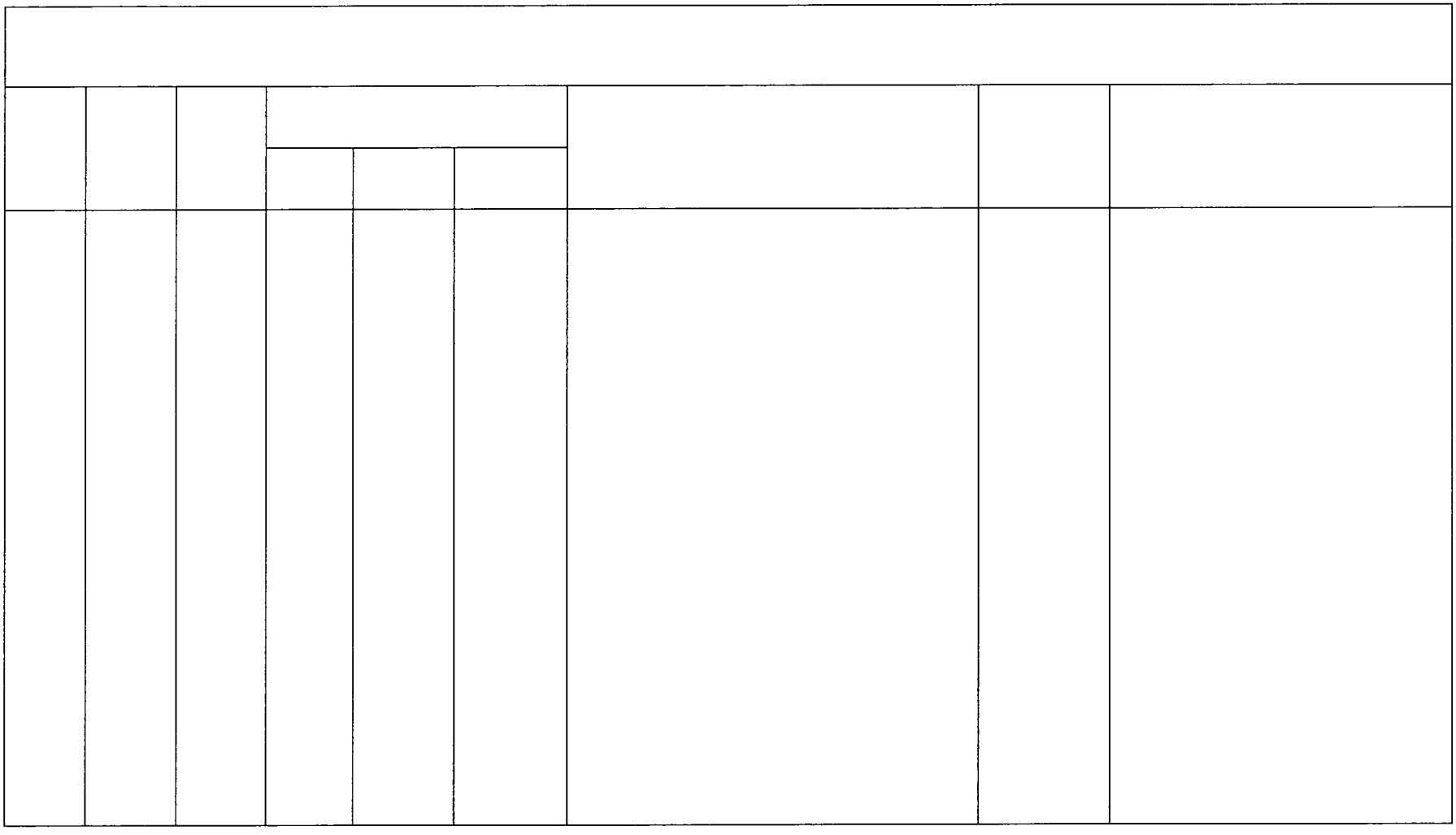
The SPE Manager is responsible for the maintenance of this Plan. This SQAP shall be assessed during the managerial review to ensure its suitability, adequacy, and effectiveness and revised to incorporate the agreed upon changes as described in Section 6.2. When improvements or deficiencies are identified, a Corrective Action Request (CAR) should be used to document the condition [[ ]]

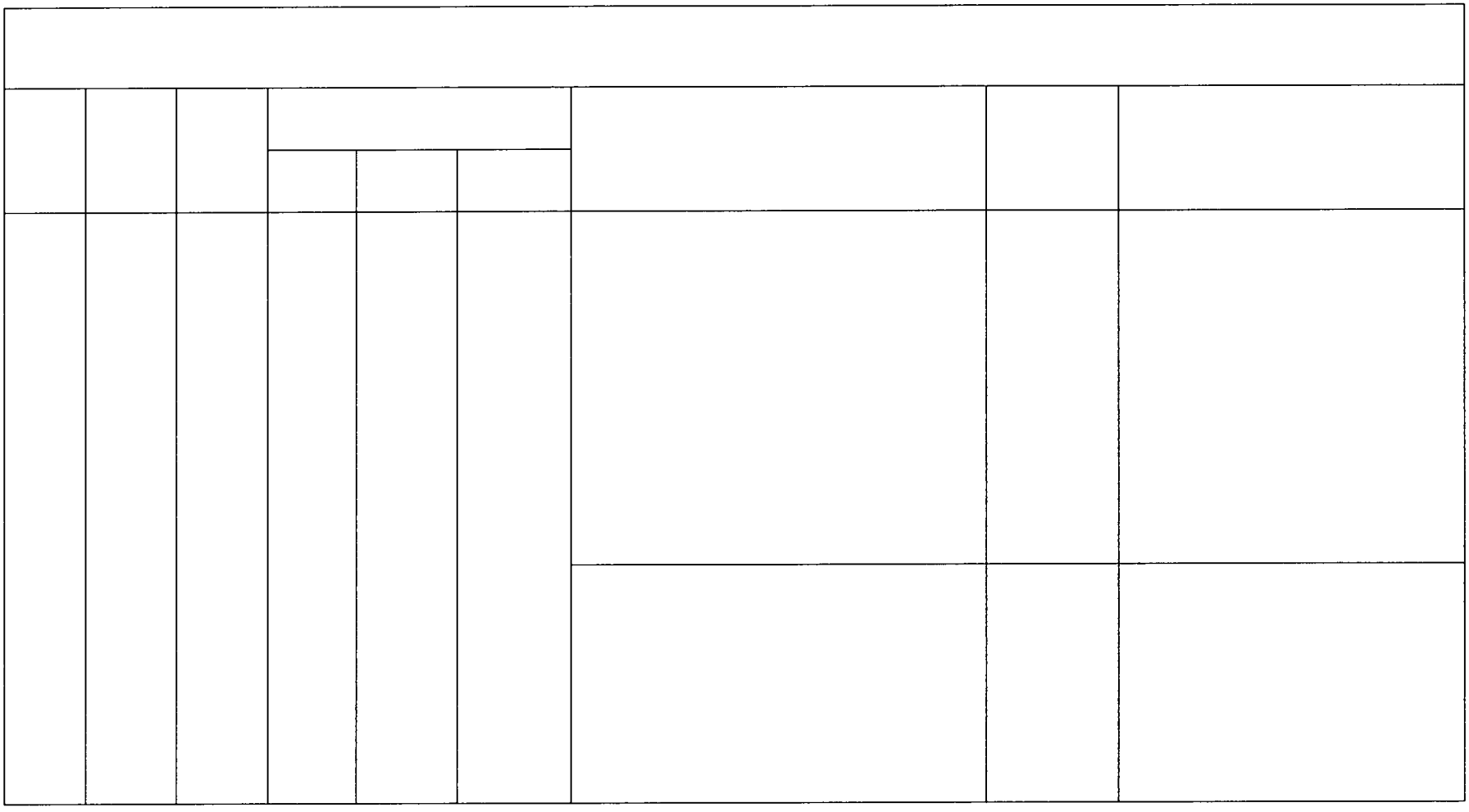
The CAR tracks activities and ensures that corrective and preventive actions are implemented. It ensures that the actions are effective in either eliminating the deficiency or improving the SQAP. If a change to the SQAP is warranted, one of the corrective activities shall determine if NRC notification is required and track the notification process as defined by the MMIS & HFE IP [2.1(1)]. The SQAP shall be revised in accordance with the Design Change Control process described in Subsection 10.4.2.2. The SPE Manager or his designated delegate shall distribute the revised SQAP to the organizations described in Section 3.0, Management Organization.

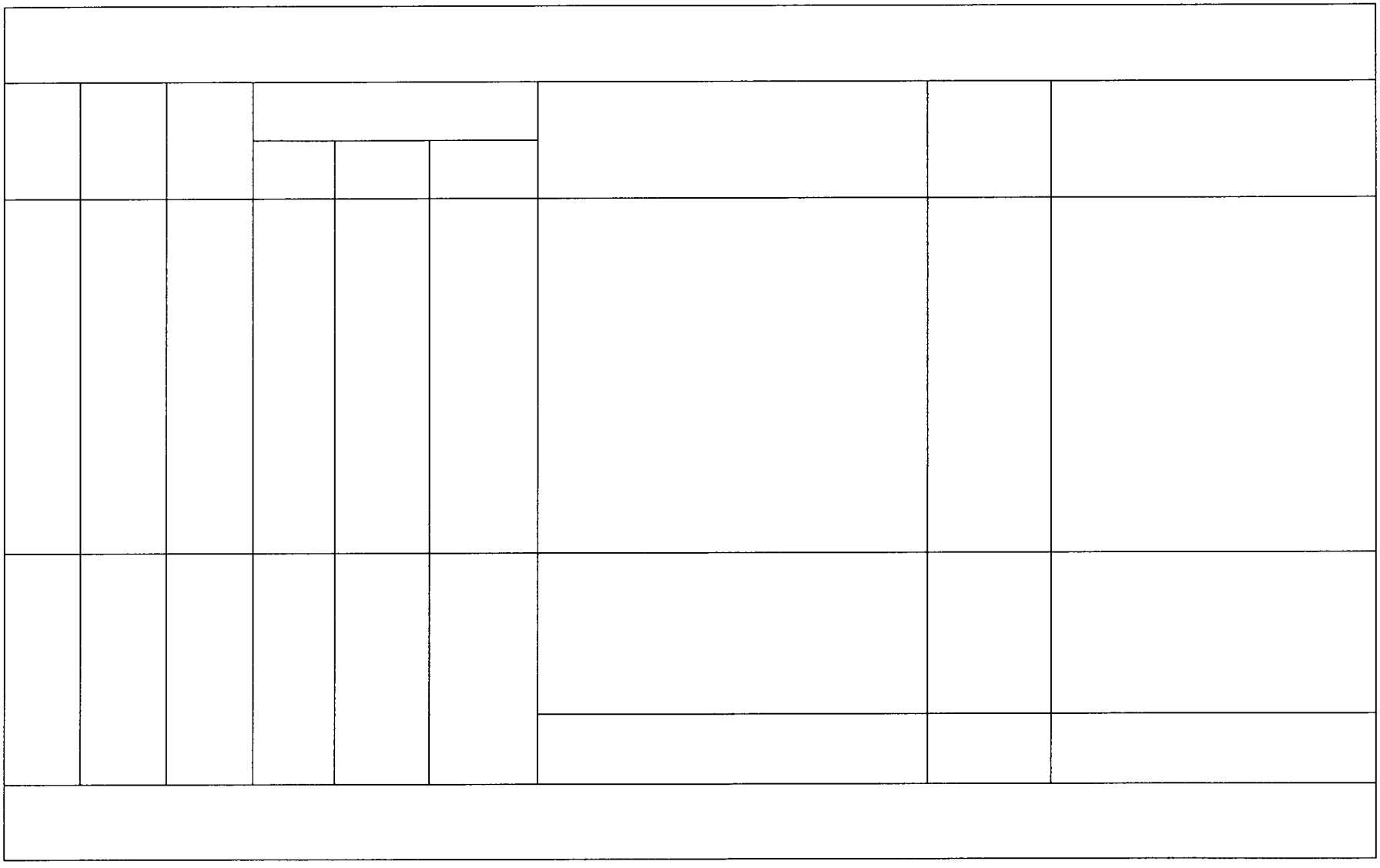
[[











								]]

APPENDIX B ACRONYMS AND ABBREVIATIONS

The following acronyms and abbreviations are used throughout this plan.

Acronym	Meaning
ADS	Automatic Depressurization System
ANSI	American National Standard Institute
AOF	Allocation of Function
ASQ	American Society for quality
ASL	Approved Suppliers List
ASME	American Society of Mechanical Engineers
ATWS	Anticipated Transients without Scram
BD	Build Description
BFM	Business Finance Manager
BR	Baseline Review
BRD	Build Release Description
BRR	Baseline Review Record
BRT	Baseline Review Team
BTP	Branch Technical Position
CAQ	Condition Adverse to Quality
CAR	Corrective Action Request
CCB	Change Control Board
CEO	Chief Executive Officer
CI	Configuration Item
CM	Configuration Management
CMM	Configuration Management Manager
CMS	Configuration Management System
CMU	Configuration Management Unit
COL	Combined Operating License

Acronym	Meaning
COTS	Commercial-Off-The-Shelf
CTS	Commitment Tracking System
DCD	Design Control Document
DLD	Detailed Logic Diagram
DRF	Design Record File
ECA	Engineering Change Authorization
ECN	Engineering Change Notice
eDRF	Electronic Design Record File
EM	Engineering Manager
EOP	Engineering Operating Procedure
EPRI	Electrical Power Research Institute
ERM	Engineering Review Memorandum
ESBWR	Economic Simplified Boiling Water Reactor
FAPCS	Fuel and Auxiliary Pools Cooling System
FAT	Factory Acceptance Test
FDDR	Field Deviation Disposition Request
FDI	Field Disposition Instruction
FMEA	Failure Modes and Effects Analysis
FRA	Functional Requirements Analysis
FX	Function
GE	General Electric Company
GEEN	GE Energy Nuclear
HFE	Human Factors Engineering
HICB	Instrumentation and Control Branch
HSI	Human System Interface
HSS	Hardware/Software Specification
I&C	Instrumentation and Control
I&C EEM	Instrumentation and Controls Electrical Engineering Manager

Acronym	Meaning
IDT	Implementation Design Team
IEEE	Institute of Electrical and Electronic Engineers
IMS	Information Management System
IP	Implementation Plan
ISO	International Standards Organization
IV&V	Independent Verification and Validation
IVVT	Independent Verification and Validation Team
LTR	Licensing Topical Report
MCR	Main Control Room
[[	]]
MMIS	Man Machine Interface System
N/A	Not Applicable
NPP	New Plant Project
NQA	Nuclear Quality Assurance
O&M	Operation and Maintenance
P&ID	Piping & Instrumentation Diagram
P&P	Policies and Procedure
PDMS	Product Data Management System
PDS	Previously Developed Software
PE	Project Engineer
PHA	Preliminary Hazards Analysis
PM	Project Manager
PMT	Project Management Team
PO	Purchase Order
POC	Point of Contact
PQC	Product Quality Certification
PRA	Probabilistic Risk Assessment
PWP	Project Work Plan

Acronym	Meaning
QA	Quality Assurance
QCE	Quality Control Engineer
RCCE	Responsible Configuration Control Engineer
RE	Responsible Engineer
RG	Regulatory Guide
RMCN	Review Memorandum Change Notice
RTA	Requirements Traceability Analysis
RTE	Responsible Test Engineer
RTM	Requirements Traceability Matrix
RTPE	Responsible Technical Project Engineer
RV	Responsible Verifier
SAE	Simulation Assisted Engineering
SAT	Site Acceptance Test
SATT	Site Acceptance Test Team
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
SDD	Software Design Description
SDP	System Development Plan
SDS	System Design Specification
[[	]]
SFRA	System Functional Requirements Analysis
SFT	Software Function Test
SFTR	Software Functional Test Report
SIntP	Software Integration Plan
SIP	Software Installation Plan
SITT	System Integrated Test Team
SMP	Software Management Plan
SOMP	Software Operations and Maintenance Plan

Acronym	Meaning
SPE	Software Project Engineering
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan
SQAT	Software Quality Assurance Team
SRP	Standard Review Plan
SRS	Software Requirements Specification
SSA	Software Safety Analysis
SSP	Software Safety Plan
SST	Software Safety Team
STrngP	Software Training Plan
SVVP	Software Validation and Verification Plan
SyAT	System Acceptance Testing
SyRS	System Requirement Specification
TA	Task Analysis
TPE	Technical Project Engineer
TSL	Training Services Lead
USNRC	United States Nuclear Regulatory Commission
V&V	Verification and Validation
VTE	Validation Test Engineer

APPENDIX C DEFINITIONS

Term	Definition
Acceptance Criteria	The criteria that a system or component must satisfy in order to be accepted by a user, customer, or other authorized entity [IEEE 610.12].
Acceptance Testing	Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system [IEEE 610.12].
Algorithm	A finite set of well-defined rules for the solution of a problem in a finite number of steps [IEEE 610.12].
Anomaly	Anything observed in the documentation or operation of software that deviates from expectations based on previously verified software products or reference documents [IEEE 610.12].
Application software	Software designed to fulfill specific needs of a user [IEEE 610.12].
Application Software Package	A collection of software modules brought together to form a single software application, e.g., an instrument (see also System Software Package and Package).
Assembly code	Computer instructions and data definitions expressed in a form that can be recognized and processed by an assembler.
Baseline	Items that have been formally reviewed and agreed upon, that thereafter serve as the basis for further development, and that can be changed only through formal change control procedures [IEEE 610.12].
Baseline Review	A formal review, conducted at the end of each process step of the software engineering design process, and requested by the Design Team's responsible TPE. The baseline review process is under the control of Software Project Engineering (SPE). The Baseline Review Team (appointed by the BRT Task Lead engineer) performs the review. These reviews are intended to confirm adherence to the project SMP and SCMP. The Baseline Reviews are performed and documented in accordance with the Software Configuration Management Plan, the Software Quality Assurance Plan, and the Software Verification and Validation Plan.
Branch testing	Testing designed to execute each outcome of each decision point in a computer program [IEEE 610.12].

Term	Definition
Build	An operational version of a system or component that incorporates a specified sub set of the capabilities that the final product will provide [IEEE 610.12].
Certification	A written guarantee that a system or component complies with its specified requirements and is acceptable for operational use [IEEE 610.12].
Code	Review of software source codes to identify coding errors and to verify that software design as specified in the Software Design Description been correctly and completely implemented.
Code review	A meeting at which software code is presented to project personnel, managers, users, customers, or other interested parties for comment or approval [IEEE 610.12].
Coding	In software engineering, the process of expressing a computer program in a programming language [IEEE 610.12].
Commitment Tracking System	System used to manage the Conditions Adverse to Quality (CAQs). A Corrective Action Request (CAR) is used to document a CAQ, or an opportunity for process/product improvement, provide for timely evaluation, and record objective evidence of actions taken. EOP 75-3.00, Self-Assessment, Corrective Action and Audits [2.3(2x)] specifies the responsibilities for actions to promptly identify, record and correct, as appropriate, CAQs, and to assure that these conditions do not affect the quality of a product or service.
Component	One of the parts that make up a system. A component may be hardware of software and may be subdivided into other components [IEEE 610.12].
Computer language	A language designed to enable humans to communicate with computers [IEEE 610.12].
Configuration control	An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification [IEEE 610.12].
Configuration Item	An aggregation of hardware, software, design documents or procedures that is designated for configuration management and treated as a single entity in the configuration management process [IEEE 610.12].

Term	Definition
Design Documentation	Design Documentation is information recorded about a specific life cycle activity. Documentation includes software life-cycle design outputs and software life cycle process documentation. A document may be in written or electronic format, and may contain text, illustrations, tables, computer files, program listings, binary images, and other forms of expression. A document for an activity may be packaged with documents for other activities, or documents for non-software life cycle activities. A document for an activity may be divided into several individual entities.
Design output	Documents, such as drawings and specifications, that define technical requirements of structures, systems, and components. For software, design outputs include the products of the development process that describe the end product that will be installed a nuclear power plant. The design outputs of a software development process include SRS, SDD, hardware and software architecture designs, code listings, system build documents, installation configuration tables, O&M manuals, and training manuals.
Design phase	The *phase* in the software life cycle during which the designs for architecture, software components, interfaces, and data are created, documented, and verified to satisfy requirements [IEEE 610.12].
Design Record File	A formal controlled information record under GEEN procedures for in-progress and completed engineering work which is retained and from which work can be retrieved.
Design Reviews	Formal, design adequacy evaluations which are performed by knowledgeable persons other than those directly responsible and accountable for the design in accordance with EOP 40-7.00. Design reviews are used to verify that product designs meet functional, contractual, safety, regulatory, industry codes and standards, and company requirements.
Deviation	A departure from a specified requirement.
Documentation	A collection of documents on a given subject [IEEE 610.12].
Error	An incorrect step, process, or data definition [IEEE 610.12].
Failure Mode and Effects Analysis	A tabular method of providing traceability from the modes by which a system may fail and the effect of that failure on the ability of the system to perform its function, or the ability of a collection of systems to recover from the failure.

Term	Definition
Fault Tree	A pictorial method of providing traceability from the modes by which a system may fail and the effect of that failure on the ability of the system to perform its function, or the ability of a collection of systems to recover from the failure.
Field Deviation Disposition Request	Field Deviation Disposition Request (FDDR) is used for documenting and disposition of the technical position for a deviation required in the field in supplied hardware, software, or services (see EOP 55-3.00).
Firmware	The combination of a hardware device and computer instructions and data that reside as read-only software on that device [IEEE 610.12].
Functional Testing	A system/software test methodology that is derived from external specifications and requirements of the system. Such testing ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions [IEEE 610.12]. Methods for functional testing include random testing and testing at boundary values. It verifies the end results at the system level, but does not check the implementation techniques, nor does it assume that all statements in the program are executed.
Implementation Phase	The *phase* in the software life cycle during which a software product is created from design documentation and debugged [IEEE 610.12].
Independent Verification and Validation (IV&V)	Verification and Validation performed by an Organization that is technically managerially and financially independent of the Organization [IEEE 610.12] and RG 1.168 Section C3 [2.2.3(1)].
Installation Phase	The *phase* in the software life cycle during which the software product is installed into its operational environment and tested to ensure that it performs as intended [IEEE 610.12].
Instrument	A hardware device used for analytical or control functions and usually containing an embedded microprocessor(s).
Integration Testing	Testing in which software elements, hardware elements, or both are combined and tested to evaluate the interaction between them [IEEE 610.12].
Interface	A shared boundary across which information is passed [IEEE 610.12].
Metric	A quantitative measure of the degree to which a system, component, or process possesses a given attribute [IEEE 610.12].

Term	Definition
Module	A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to, or output from, and assembler, compiler, linkage editor, or executive routine [IEEE 610.12].
Operations and Maintenance Phase	The *phase* in the software life cycle during which the software product is functioning in its operational environment, monitored for satisfactory performance and modified as necessary to correct problems or to respond to changing requirements [IEEE 610.12].
Package	A separately compliable software component consisting of related data types, data objects and sub-programs [IEEE 610.12].
Path Testing	Testing designed to execute all or selected paths through a computer program [IEEE 610.12].
Planning Phase	The initial *phase* of a software development project, in which project scope, purpose, strategy, schedule and milestones are established and user needs through documentation (for example, system definition documentation and procedures) are described and evaluated.
Procedure	A course of action to be taken to perform a given task [IEEE 610.12].
Process	A sequence of steps performed for a given purpose, e.g., the software development process [IEEE 610.12].
Project *Management* Plan	A document that describes the technical and management approach to be followed for a project. The plan typically describes the work to be done, the resources required, the methods to be used, the procedures to be followed, the schedules to be met, and the way that the project will be organized [IEEE 610.12].
Regression Testing	Selective re-testing of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements [IEEE 610.12].

Term	Definition
Requirement	A condition or capability that must be met or possessed by a system or system component to satisfy a contract standard specification or other formally imposed documents [IEEE 610.12]. In specifying requirements, the word **shall** is used to indicate mandatory requirements and from which no deviation is permitted **('shall'** and **'required to'** are equivalent in meaning**)**. Requirements are not specified with the word **should.** Instead, it is used to indicate that a recommended course of action and is particularly suitable, without mentioning or excluding other courses of action; Also, a certain course of action is preferred but not necessarily required; Also, that (in the negative form) a certain course of action is not prohibited ('should' and 'recommended' are equivalent in meaning).
Requirements Phase	The *phase* in the software life cycle during which the requirements for a software product are defined and documented [IEEE 610.12].
Requirements *Traceability* Analysis	The process of studying user needs to arrive at a definition of system, hardware, or software requirements [IEEE 610.12].
Responsible Configuration Control Engineer	The person assigned responsibility for the configuration management of the I&C software products.
Responsible Engineer	The person responsible for a given technical item, e.g., the design and development of the documentation.
Responsible Technical Project Engineer	The person with overall technical responsibility for ensuring that the hardware and software design of a software product meets the specified requirements.
Responsible Verifier	The Responsible Verifier(s) is an individual who has the independence described in EOP 42-6.00 for verifications, or in EOP 42-6.10 for deferred verifications of design process and the accompanying documents.
Retirement	Permanent removal of a system or component from its operational environment [IEEE 610.12].
Simulation	A model that behaves or operates like a given system when provided a set of controlled inputs [IEEE 610.12].
Software Class N2	Nonsafety-related system software whose failure cannot adversely affect a safety related function.

Term	Definition
Software Class N3	Nonsafety-related systems software whose failure could challenge safety systems as defined below: a. Software whose inadvertent response to stimuli, failure to respond when required, response out-of-sequence could directly result in a accident or transient as defined in the DCD, chapter 15 [2.1(6)]. b. Software that is intended to mitigate the result of an accident. c. Software that is intended to recover from the result of an accident.
Software Class Q	Software performs functions classified per EOP 65-2.10 as Safety-Related.
Software Development Process	The process by which user needs are translated into a software product. The process involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, testing the code, and sometimes, installing and checking out the software for operational use [IEEE 610.12].
Software Feature	A distinguishing characteristic of a software item, such as, performance, portability, or functionality.
Software Item	Source code, object code, job control code, control data, or a collection of these items [IEEE 610.12].
Software Life cycle	The period of time that begins when a software product is conceived and ends when the software is no longer available for use [IEEE 610.12].
Software Life cycle Phase	The division of the software life cycle into discrete logical units. The I&C software life cycle is divided into eight *phases*, namely, Planning, Requirements, Design, Implementation, Integration, Validation, Installation, and Operation & Maintenance.
Software Module	See Module
Software Package	See Package
Software Unit	See Module
Source Code	Computer instructions and data definitions expressed in a form suitable for input to an assembler, compiler, or other translator.
Statement testing	Testing designed to execute each statement or a computer program [IEEE 610.12].
Stress testing	Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements [IEEE 610.12].

Term	Definition
Supplemental Document	Controlled documents that are referenced or used in conjunction with this plan. These are the enabling documents that either augment or enable the performance of the activities stated in this plan.
Support software	Software that aids in the development or maintenance of other software; for example, compilers, loaders, and other utilities [IEEE 610.12].
Supporting Document	Controlled documents used in the production of this plan. These documents form the design basis for the activities stated in this plan.
System Testing	Testing conducted on a complete, integrated system to evaluate the systems compliance with its specified requirements [IEEE 610.12].
Technical Project Engineer	The person with overall technical responsibility for ensuring that the hardware and software design of a software product meets the specified requirements.
Test case	A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement [IEEE 610.12].
Test Item	A software item that is an object of testing [IEEE 610.12].
Test Log	A chronological record of all relevant details about the execution of a test [IEEE 610.12].
Test Objective	An identified set of software features to be measured under specified conditions by comparing actual behavior with the required behavior described in the software documentation [IEEE 610.12].
Test Phase	The *phase* in the software life cycle during which the components of a software product are integrated with the hardware and evaluated to determine whether or not performance requirements have been satisfied [IEEE 610.12].
Test Plan	A document describing the scope, approach, resources, and schedule of intended test activities. It identifies test items, the features to be tested, the testing tasks, who will do such task, and any risks requiring contingency planning [IEEE 610.12].
Traceability Matrix	A matrix that records the relationship between two or more product specifications (i.e., design documentation) of the development process (e.g., a matrix that records the relationship between the requirements and the design of a given software component) [IEEE 610.12].

Term	Definition
Unit *Module* Testing	Testing of individual hardware or software units or groups of related units [IEEE 610.12].
User interface	An interface that enables information to be passed between a human user and hardware or software components of a computer system [IEEE 610.12].
Verification and Validation (V&V)	The design verification activities performed in accordance with GEEN EOPs 40-7.00 (Design Reviews) or 42-6.00 (Independent Design Verification) based on 10CFR50 Appendix B [2.2.2(1)] or equivalent to ensure the quality of the design process and the associated documents produced. For Software Class Q software products, the verification and validation activities are performed by the SPE in accordance with the design process (SVVP) to ensure the quality of the associated documents produced.

APPENDIX D SOFTWARE CHARACTERISTICS

Software characteristics important to safety system software as defined by NUREG 0800, SRP, [2.2.1(1)]. These characteristics are divided into two sets:

- Software functional characteristics
- Software development process characteristics

Term	Definition
Accuracy	The degree of freedom from error of sensor and operator input, the degree of exactness exhibited by an approximation or measurement, and the degree of freedom from error of actuator output.
Functionality	The operations, which must be carried out by the software. Functions generally transform input information into output information in order to affect the reactor operation. Inputs may be obtained from sensors, operators, other equipment, or other software. Outputs may be directed to actuators, operators, other equipment, or other software.
Reliability	The degree to which a software system or component operates without failure. This definition does not consider the consequences of failure, only the existence of failure.
Robustness	The ability of a software system or component to function correctly in the presence of invalid inputs or stressful environmental conditions. This includes the ability to function correctly despite some violation of the assumptions in its specification.
Safety	Those properties and characteristics of the software system that directly affect or interact with system safety considerations. The other characteristics discussed in Chapter 7 of NUREG 0800 SRP [2.2.1(1)] are important contributors to the overall safety of the software-controlled safety system, but are primarily concerned with the internal operation of the software. The safety characteristic, however, is primarily concerned with the effect of the software on system hazards and the measures taken to control those hazards.
Security	The ability to prevent unauthorized, undesired, and unsafe intrusions. Security is a safety concern insofar as such intrusions can affect the safety-related functions of the software.
Timing	The ability of the software system to achieve its timing objectives within the hardware constraints imposed by the computing system being used.

Term	Definition
Completeness	Those attributes of the planning documents, implementation process documents and design outputs that provide full implementation of the functions required of the software. The functions, which the software is required to perform are derived from the general functional requirements of the safety system, and the assignment of functional requirements to the software in the overall system design.
Consistency	The degree of freedom from contradiction among the different documents and components of a software system. There are two aspects to consistency. Internal consistency denotes the consistency within the different parts of a component for example, a software design is internally consistent if no set of design elements are mutually contradictory. External consistency denotes the consistency between one component and another for example, software requirements and the resulting code are consistent with one another if there are no contradictions between the requirements and the code.
Correctness	The degree to which a design output is free from faults in its specification, design, and implementation. There is considerable overlap between correctness properties and properties of other characteristics such as accuracy and completeness.
Style	The form and structure of a planning document, implementation process document or design output. Document style refers to the structure and form of a document. This has connotations of understandability, readability, and modifiability. Programming style refers to the programming language characteristics of the software and programming techniques, which are mandated, encouraged, discouraged, or prohibited in a given implementation.
Traceability	The degree to which each element of one life cycle product can be traced forward to one or more elements of a successor life cycle product, and can be traced backwards to one or more elements of a predecessor life cycle product.
Unambiguity	The degree to which each element of a product, and of all elements taken together, have only one interpretation.
Verifiability	The degree to which a software planning document, implementation process document or design output is stated or provided in such a way as to facilitate the establishment of verification criteria and the performance of analyses, reviews, or tests to determine whether those criteria have been met.

APPENDIX E V&V TASKS DEFINITIONS

Term	Definition
Criticality Analysis	A structured evaluation of the software characteristics (e.g., safety, security, complexity, performance) for severity of impact of system failure, system degradation, or failure to meet software requirements or system objectives [IEEE-1012].
Traceability Analysis	Trace the software requirements (SRS and HHS) to system requirements (SDS) and system requirements to software requirements. Analyze identified relationships to correctness, consistency, completeness, and accuracy [IEEE-1012].
Interface Analysis	Verify and validate that the requirements for software interfaces with hardware, user operator and other systems are correct, consistent, complete, accurate, and testable [IEEE-1012].
Hazard Analysis	A systematic qualitative or quantitative evaluation of software for undesirable outcomes resulting from the development or operation of a system. These outcomes may include injury, illness, death, mission failure, economic loss, property loss, environmental loss, or adverse social impact. This evaluation may include screening or analysis methods to categorize, eliminate, reduce, or mitigate hazards [IEEE-1012].
Risk Analysis	The systematic use of available information to identify hazards and estimate the risk to individuals or populations, property or the environment [IEEE-1012 Annex I].
Algorithm Analysis	Verify the correct implementation of algorithms. Equations, mathematical formulations, or expressions. Rederive any significant algorithms, and equations from basic principles and theories. Compare against established references or proven past historical data. Validate the algorithms, equations, mathematical formulations, or expressions with respect to the system and software requirements. Ensure that the algorithms and equations are appropriate for the problem solution. Validate the correctness of any constraints or limitations such as rounding, truncation, expression simplifications, best-fit estimations, non-linear solutions imposed by the algorithms and equations [IEEE-1012].

Term	Definition
Control Flow Analysis	Assess the correctness of the software by diagramming the logical control. Examine the flow of the logic to identify missing, incomplete, or inaccurate requirements. Validate whether the flow of control amongst the functions represents a correct solution to the problem [IEEE-1012].
Database Analysis	Evaluate database design as part of a design review process could include the following: **Physical Limitations Analysis** Identify the physical limitations of the database such as maximum number of records, maximum record length, largest numeric value, smallest numeric value, and maximum array length in a data structure and compare them to designed values. **Index vs. Storage Analysis** Analyze the use of multiple indexes compared to the volume of stored data to determine if the proposed approach meets the requirements for data retrieval performance and size constraints. **Data Structures Analysis** Some database management systems have specific data structures within a record such as arrays, tables, and date formats. Review the use of these structures for potential impact on requirements for data storage and retrieval. **Backup and Disaster Recovery Analysis** Review the methods employed for backup against the requirements for data recovery and system disaster recovery and identify deficiencies [IEEE-1012].

Term	Definition
Data Flow Analysis	Evaluate data flow diagrams as part of a design review process. This could include the following: **Symbology Consistency Check.** The various methods used to depict data flow diagrams employ very specific symbology to represent the actions performed. Verify that each symbol is used consistently. **Flow Balancing.** Compare the output data from each process block to the data inputs and the data derived within the process to ensure that data is available when required. This process does not specifically examine timing of sequence considerations. **Confirmation of Derived Data.** Examine the data derived within a process for correctness and format. Data designed to be entered into a process by operator action should be confirmed to ensure availability. **Keys to Index Comparison.** Compare the data keys used to retrieve data from data stores within a process to the database index design to confirm that no invalid keys have been used and the uniqueness properties are consistent [IEEE-1012].
Simulation Analysis	Use a simulation to exercise the software or portions of the software to measure the performance of the software against predefined conditions and events. The simulation can take the form of a manual walkthrough of the software against specific program values and inputs. The simulation can also be another software program that provides the inputs and simulation of the environment to the software under examination. Simulation analysis is used to examine critical performance and response time requirements or the software's response to abnormal events and conditions [IEEE-1012].
Sizing and timing Analysis	Collect and analyze data about the software functions and resource utilization to determine if system and software requirements for speed and capacity are satisfied. The types of software functions and resource utilization 'issues include, but are not limited to the following: • CPU load. • Random access memory and secondary storage (e.g., disk, tape) utilization. • Network speed and capacity. • Input and output speed. • Sizing and timing analysis is started at software design and iterated through acceptance testing [IEEE-1012].

Term	Definition
Software Regression Analysis and Testing	Determine the extent of V&V analysis and tests that must be repeated when changes are repeated when changes are made to any previously examined software products. Assess the nature of the change to determine ripple or side effects and impacts on other aspects of the system. Rerun test cases based on changes, error corrections, and impact assessment, to determine errors spawned by software modifications. [IEEE 1012 Annex G]
Security Assessment	Evaluate the security controls on the system to ensure that they protect the hardware and software components from unauthorized use, modifications, and disclosures, and to verify the accountability of the authorized users. Verify that these controls are appropriate for achieving the system's security objectives. A system security assessment should include both the physical components (e.g., computers, controllers, networks, modems, radio frequency, infrared devices) and logical components (e.g., operating systems, utilities, application programs, communication protocols, data, administrative operating policies and procedures). [IEEE-1012]
Disaster Recovery Plan Assessment	Verify that the disaster recovery plan is adequate to restore critical operation of the system in the case of an extended system outage. The disaster recovery plan should include the following: • Identification of the disaster recovery team and a contact list. • Recovery operation procedures. • Procedure for establishing an alternative site including voice and data communications, mail, and support equipment. • Plans for replacement of computer equipment • Establishment of a system backup schedule • Procedures for storage and retrieval of software, data, documentation, and vital records off-site. • Logistics of moving staff, data, documentation, etc [IEEE-1012].
Distributed Architecture Assessment	Assess the distribution of data and processes in the proposed architecture for feasibility, timing compliance, availability of telecommunications, cost, backup and restore features, downtime, system degradation, and provisions for installation of software updates [IEEE-1012].

Term	Definition
Independent Risk Assessment	Conduct an independent risk assessment on any aspect of the software project and report on the findings. Such risk assessments will be primarily from a system perspective. Examples of risk assessment include appropriateness of the selected development methodology or tools for the project; and quality risks associated with proposed development schedule alternatives [IEEE-1012].
Anomaly Evaluation	Assessment of software that deviates from documented requirements, specifications, design, user documents, or standards. The assessment should include risk based on probability and severity of occurrence. [IEEE-1012].
Evaluation of New Constraints	Evaluate new constraints (e.g., operational requirements, platform characteristics, operating environment) on the system or software requirements to verify the applicability of the SVVP. Software changes are maintenance activities [IEEE-1012].
System Software Assessment	Assess system software (e.g., operating system, computer-aided software engineering tools, data base management system, repository, telecommunications software, graphical user interface) for feasibility, impact on performance and functional requirements, maturity, supportability, adherence to standards, developer's knowledge of and experience with the system software and hardware, and software interface requirements [IEEE-1012].
Previously Developed Software Assessment	Only formal assessments of existing software will be addressed. Assessment of existing software is an iterative assessment (the comparison of the software in the same domain over time or comparative to other domains within the same existing software being studied). The assessment can be formative, summative, objective, subjective, criterion-referenced, and/or norm-referenced.
Operation Procedures Evaluation	Verify that the operating procedures are consistent with the user documentation and conform to the system requirements.
Software V&V plan (SVVP) generation	Generate and SVVP for all life cycle processes. The SVVP MAY require updating throughout the life cycle. Outputs of other activities are inputs to the SVVP. Establish a baseline SVVP prior to the requirements V&V activities. Identify project milestones in the SVVP. Schedule V&V tasks to supports project managements reviews and technical reviews [IEEE-1012].

Term	Definition
Audit Performance	Provide an independent assessment of whether a software process and its products conform to applicable regulations, standards, plans, procedures, specifications and guidelines. Audits may be applied to any software process or product at any development stage. Audits may be initiated by the supplier, the acquirer, the developer or other involved party such as a regulatory agency. The initiator of the audit selects the audit team and determines the degree of independence required. The initiator of the audit and the audit team leader establish the purpose, scope, plan, and reporting requirements for the audit. The auditors collect sufficient evidence to decide whether the software processes and products meet the evaluation criteria. They identify major deviations, assess risk to quality, schedule, and cost and report their findings. Examples of processes that could be audited include configuration management practices, use of software tools, degree of integration of the various software engineering disciplines particularly in developing architecture, security issues, training, project management [IEEE-1012].
Concept Documentation evaluation	Verify that the concept documentation satisfies user needs and is consistent with acquisition needs. Validate constraints of interfacing systems and constraints or limitations of proposed approach [IEEE-1012].
Planning the interface between the V&V effort and supplier	Plan the V&V schedule for each V&V task. Identify the preliminary list of development processes and products to be evaluated by the V&V processes. Describe V&V access rights to proprietary and classified information. It is recommended that he plan be coordinated with the acquirer. Incorporate the project software integrity level scheme into the planning process [IEEE-1012].
Software requirements evaluation	Evaluation of the essential requirements (i.e., functions, performance, design constraints, and attributes) of the software.
Support tool evaluation	The systematic determination of merit, worth, and significance of a programming tool. Support tool is a program or application used to create, debug, or maintain other programs and applications.
Software design evaluation	Evaluate the design elements (SDD) for correctness, consistency, completeness, accuracy, readability, and testability [IEEE-1012].
Source code and source code documentation evaluation	Evaluate the source code components (Source Code Documentation) for correctness, consistency, completeness, accuracy, readability, and testability [IEEE-1012].

Term	Definition
Installation checkout Report Evaluation	Conduct analyses or test to verify that the installed software corresponds to the software subjected to V&V. Verify that the software code and databases initialize, execute, and terminate as specified. In the transition from one version of software to the next, the V&V efforts shall validate that the software can be removed from system to without affecting the functionality of the remaining system components. The V&V effort shall verity the requirements for continuous operation and service during transition, including user notification.
Installation configuration audit	Verify that all software products required to correctly install and operate the software are present in the installation package. Validated that all site dependent parameters or conditions to verify supplied values are correct.
Planning Phase Inspection	Planning Phase Baseline Review
Requirements Phase Inspection	Requirements Phase Baseline Review
Design Phase Inspection	Design Phase Baseline Review
Implementation Phase Inspection	Implementation Phase Baseline Review
Test Phase Inspection	Test Phase Baseline Review
Test Certification	Certify the test results by verifying that the tests were conducted using baselined requirements, a configuration control process, and repeatable tests, and by witnessing the tests. Certification may be accomplished at a software configuration item level or at a system level [IEEE-1012].
Test Witnessing	Monitor the fidelity of test execution to the specified test procedures, and witness the recording of test results. When a test failure occurs, the testing process can be continued by 1) implementing a "workaround" to the failure; 2) inserting a temporary code patch; or **3)** halting the testing process and implementing a software repair. In all cases, assess the test continuation process for test process breakage (e.g., some software is not tested or a patch is left in place permanently), adverse impact on other, tests and loss of configuration control. Regression testing should be done for all the software affected by the test failure [IEEE-1012].

Term	Definition
Project Management Oversight Support	Assess project development status for technical and management issues, risks, and problems. Coordinate oversight assessment with the acquirer and development organization. Evaluate project plans, schedules, development processes, and status. Collect, analyze, and report on key project metrics [IEEE-1012].

Figure 4. Baseline Review Record

This is an example of the form to be used for the Baseline Review Record.

PLANNING BASELINE REVIEW RECORD

1st BASELINE

Revision 0

PROJECT:		
PRODUCT:		DATE:

CONFIGURATION MANAGEMENT:	
OBJECTIVES:	
SCOPE:	
ITEMS TO BE BASELINED:	APPROVED DATE:
1.	
2.	
3.	

V&V AND SSA SUMMARY:
ASSESSNENT:
RECOMMENDATION:

BASELINE REVIEW TEAM MEMBERS:
COMMENTS:
CONCLUSION:

Baseline Approved By Baseline Review Team Task Lead: _____

[Sign, date and Print Name]

Figure 5. Software Library Structure

Software library structure is dependant upon the medium and location of the library. Several software libraries for a single project may be required due to different media requirements or because of the use of COTS software or PDS. The following is an example of a structure of a Software Library located on a VAX development platform:

Directory Structure: [*xxxxx*.bbb.ccc]

where:

Extension	**Example**
xxxxx is the Product Type	PRM
bbb is the Category	BRR - Baseline Review Record SOURCECODE - Source Code etc.
ccc is the Released Software Revision	REV0 - Initial Software Release, REV1 - First Revision, etc.

For example:

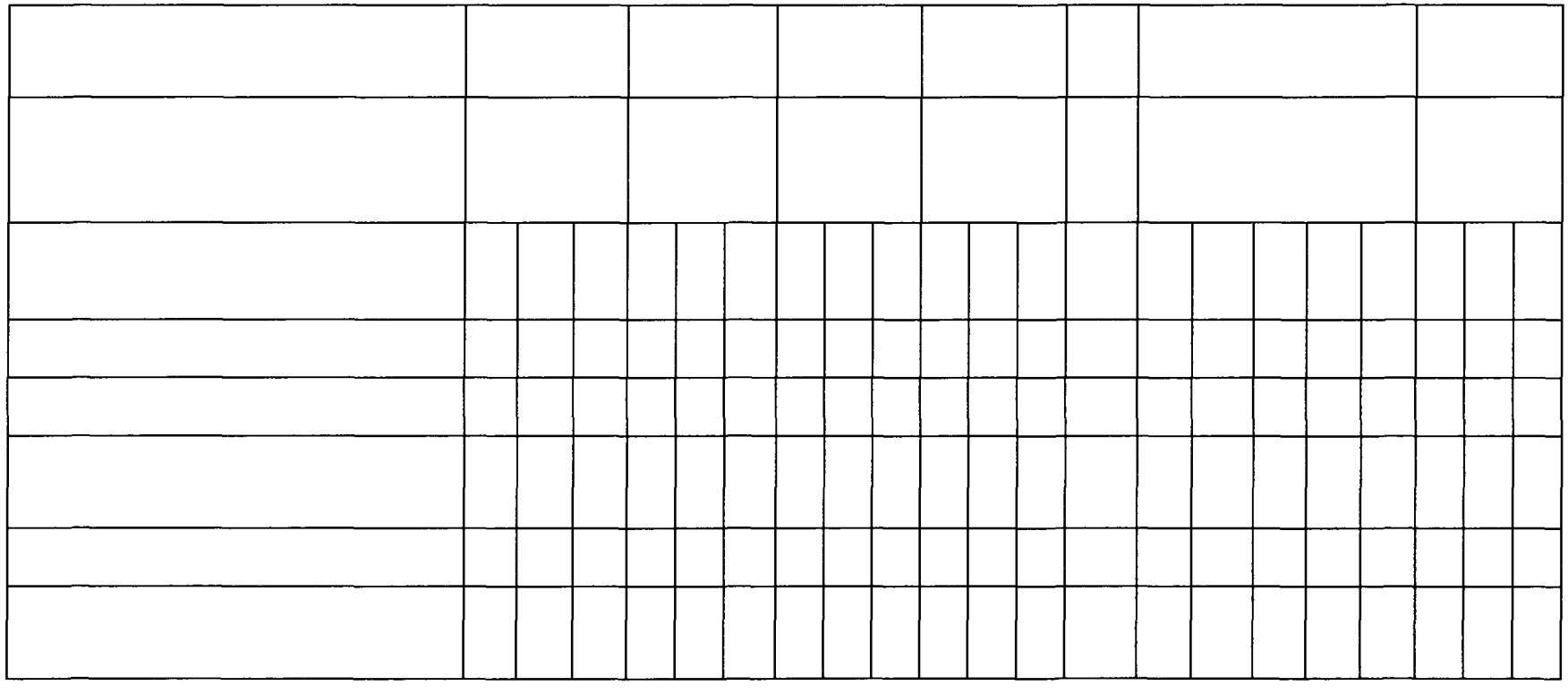
PRM.SOURCECODE.REV0 is the directory location of Revision 0 of the Software Source code for the NUMAC Process Radiation Monitor (PRM).

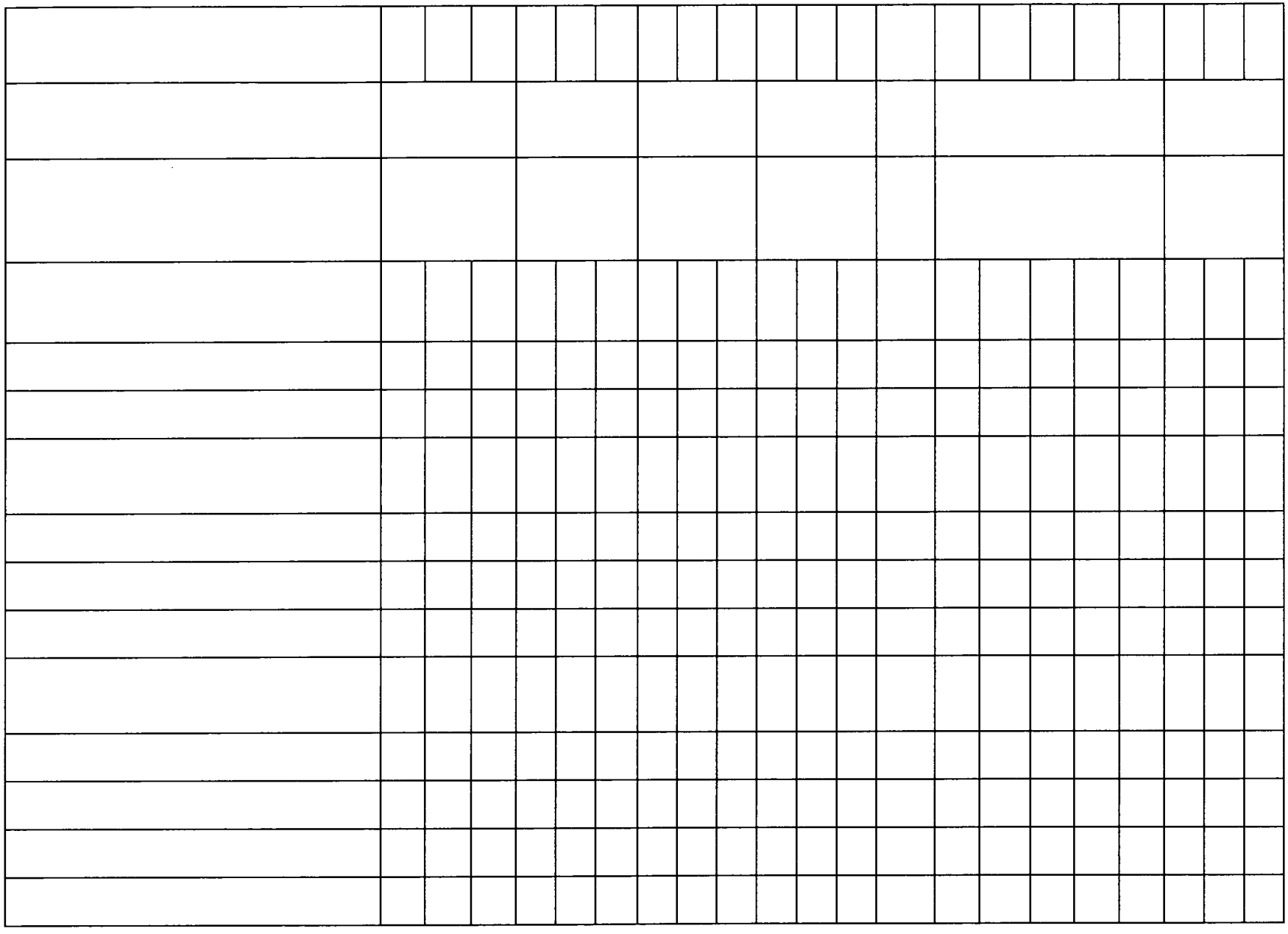
For each software library used in the project, a supplemental document defining the software library structure shall be generated, stored in a DRF and linked to each appropriate DRF.

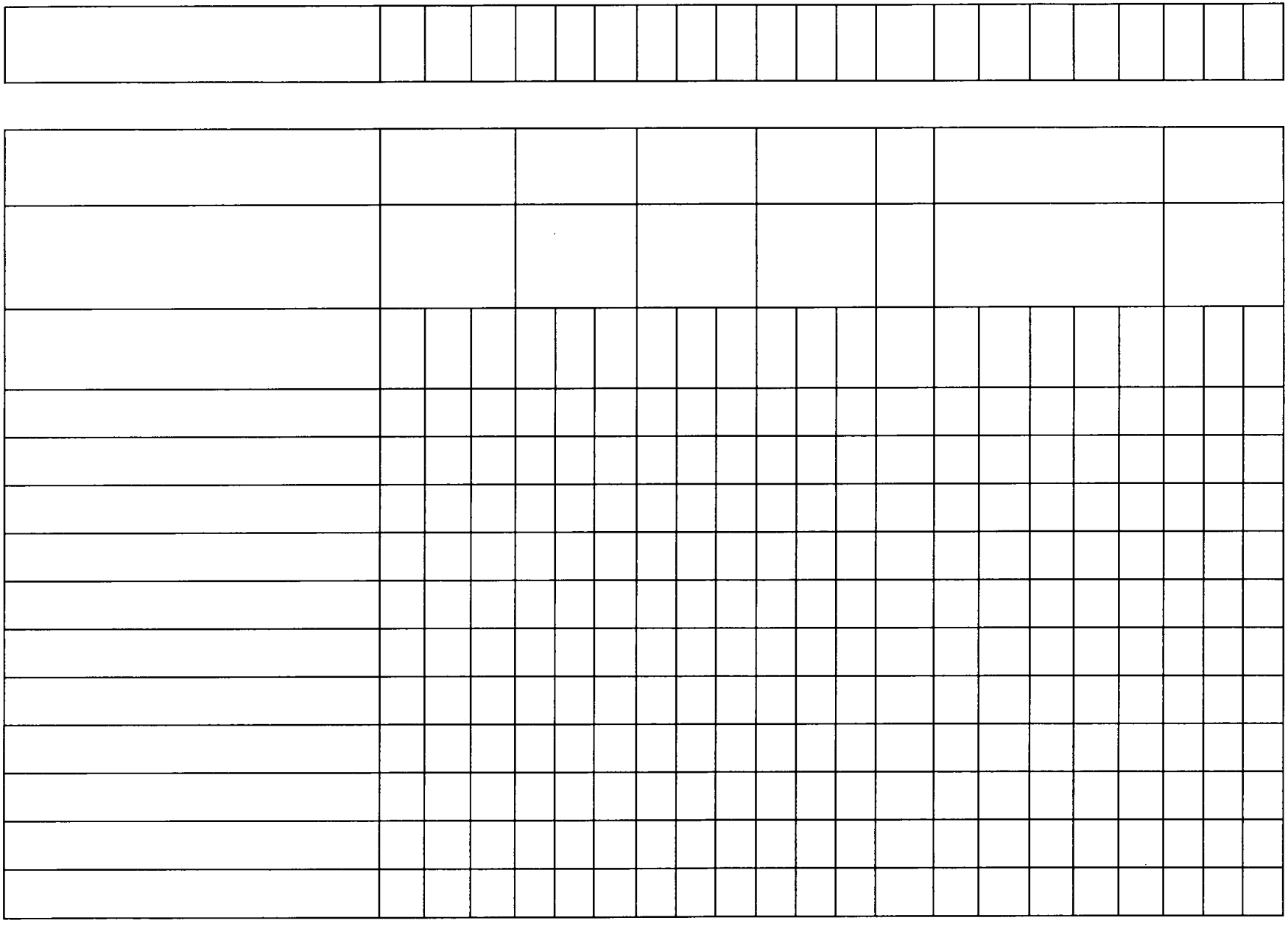
[[

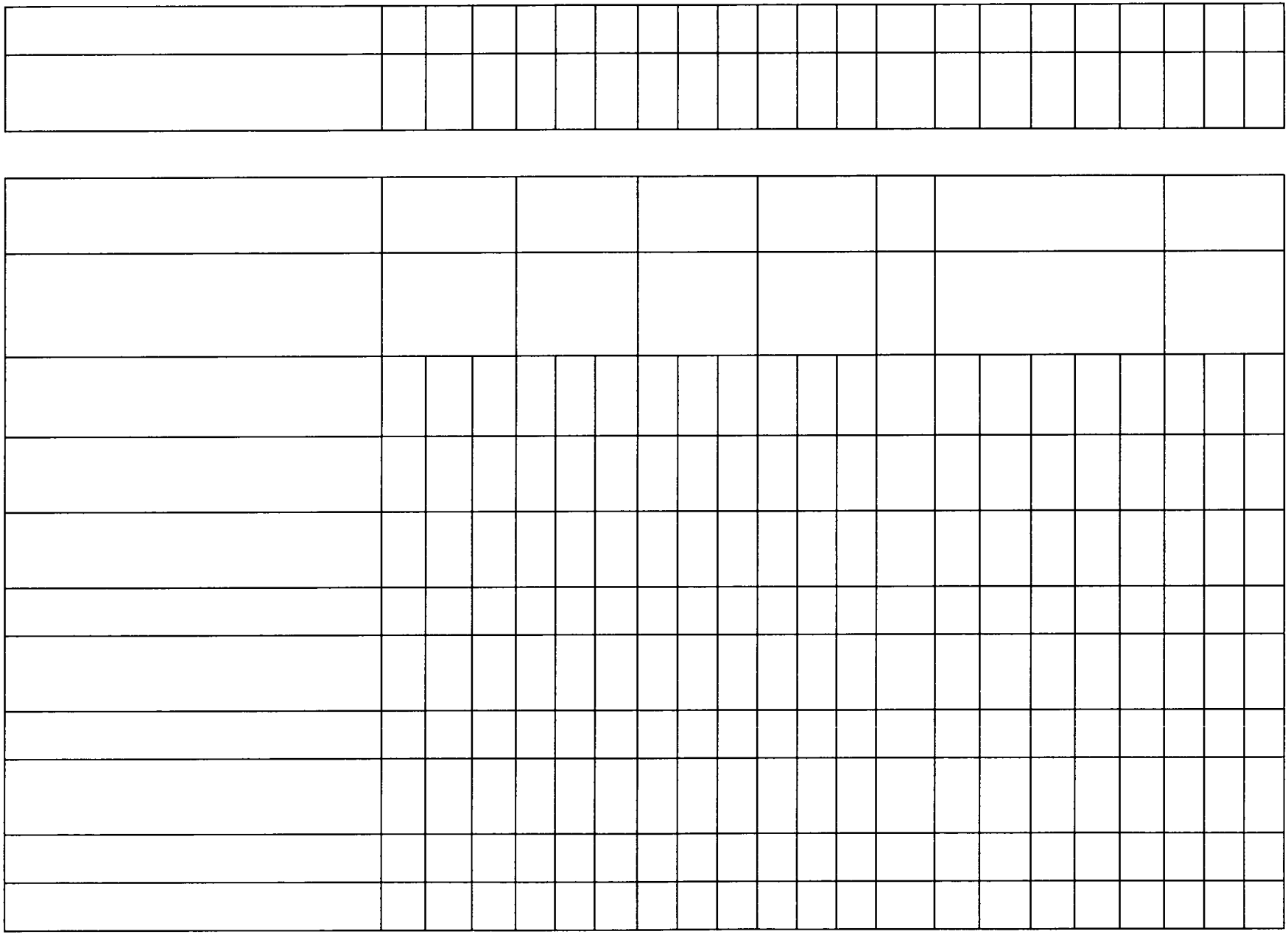
					]]

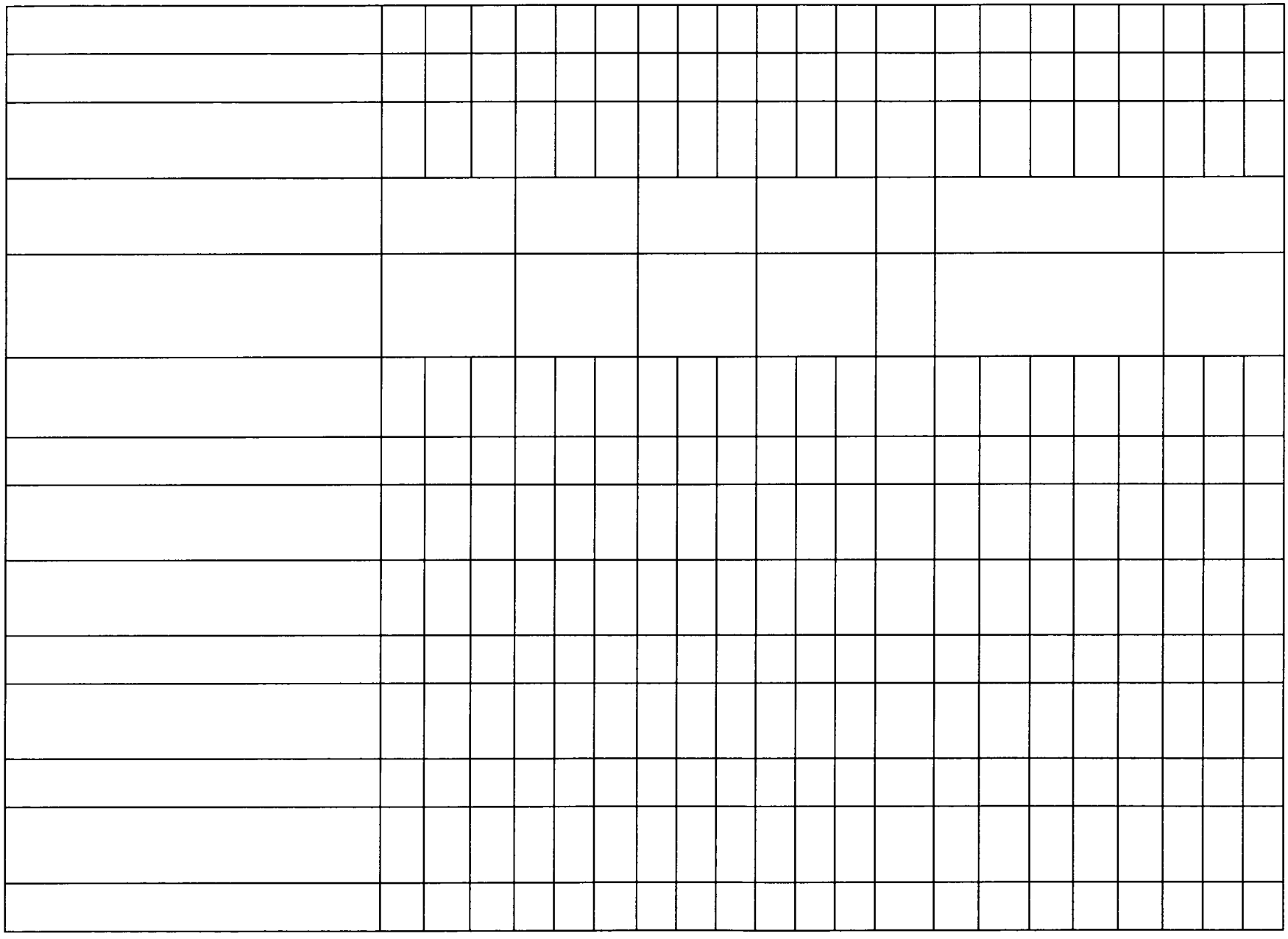
[[

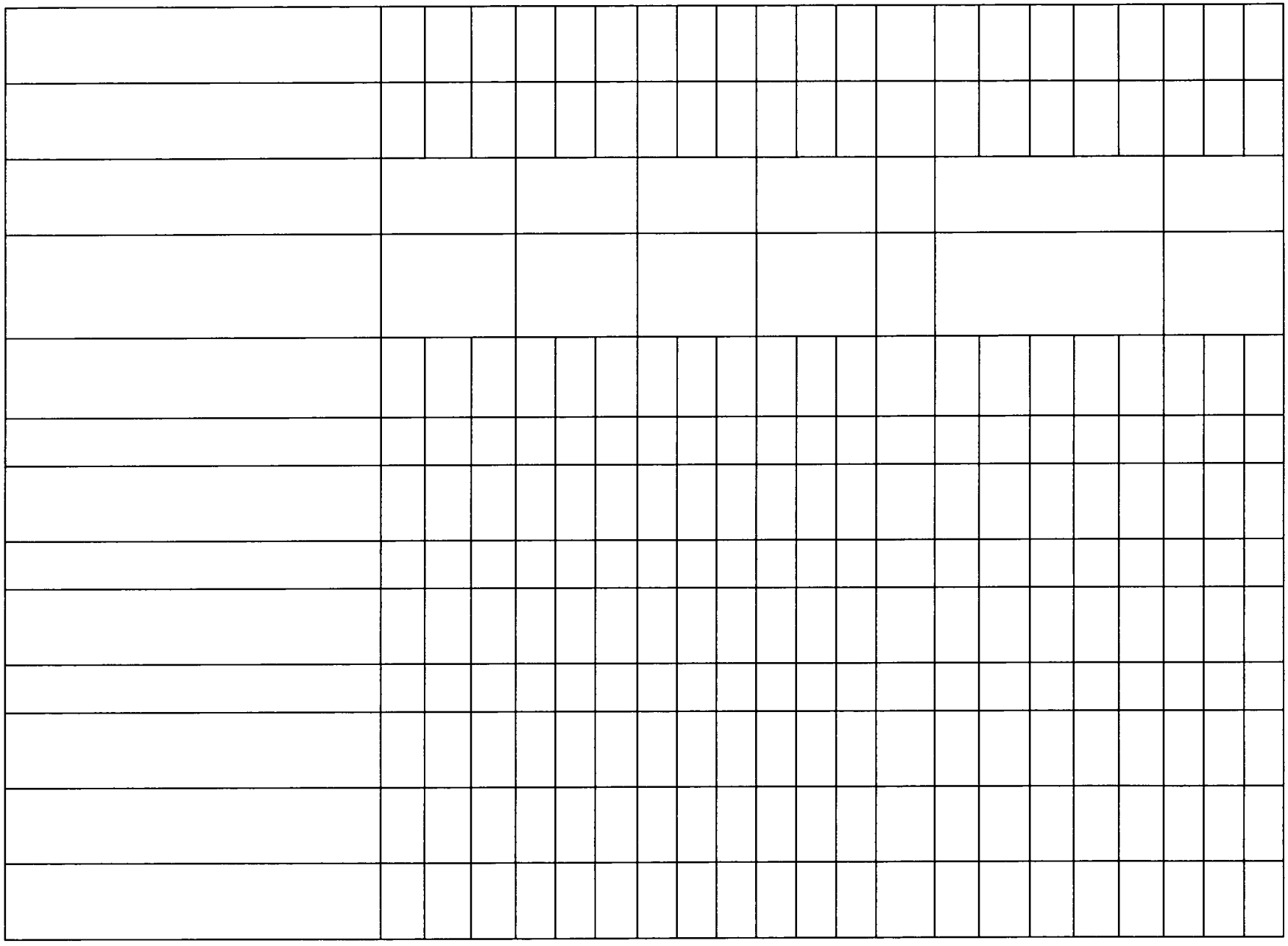


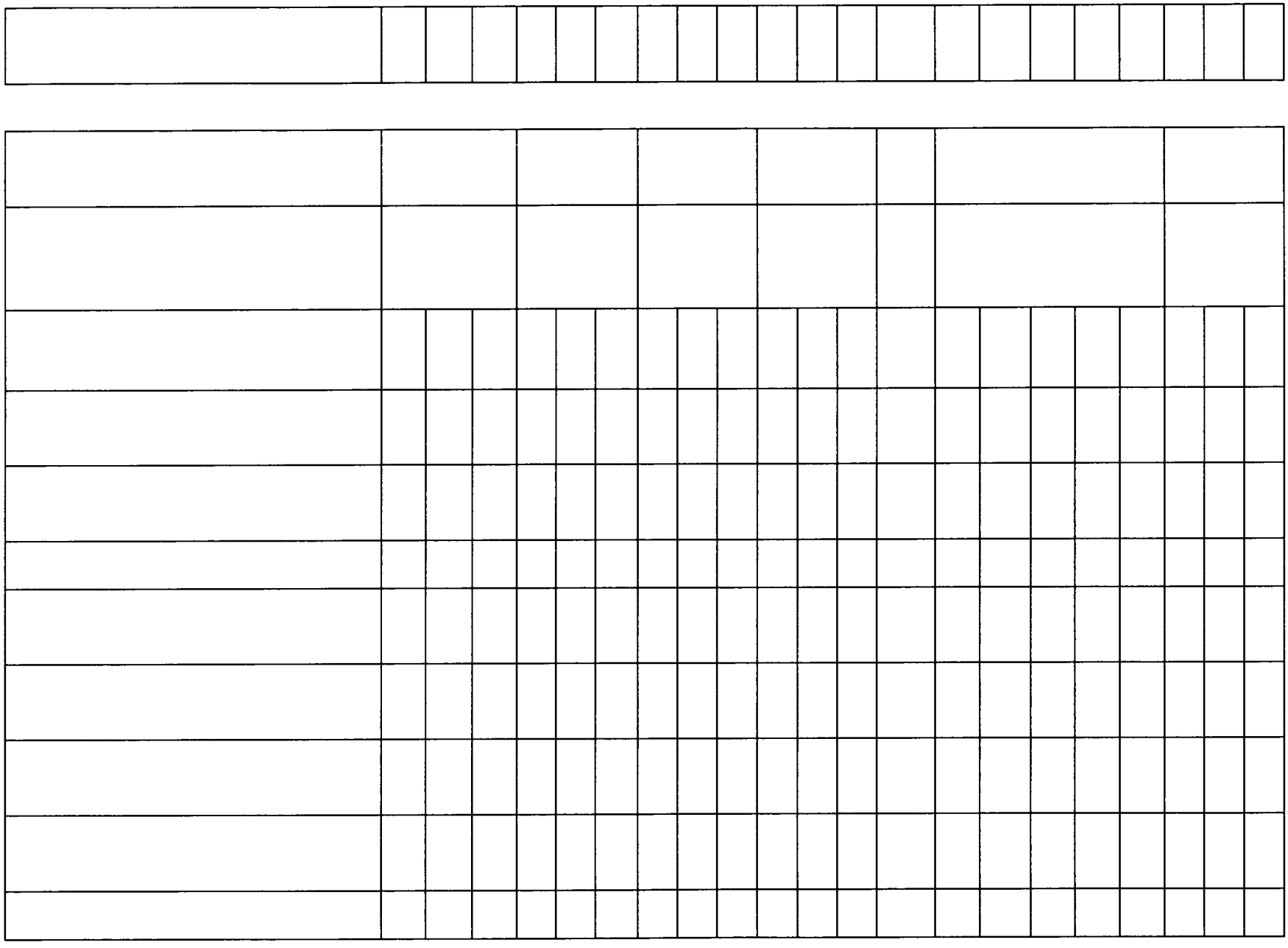


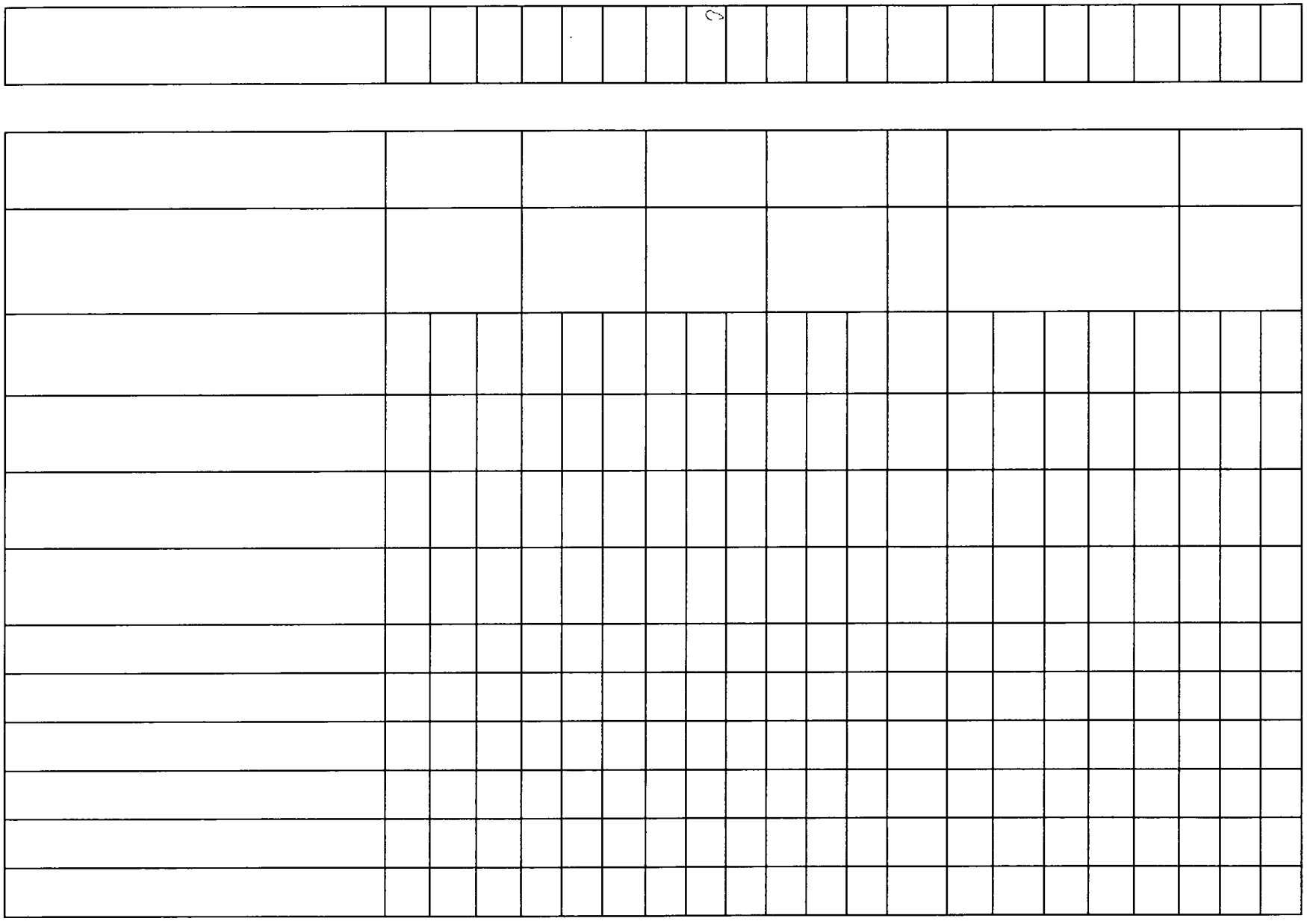


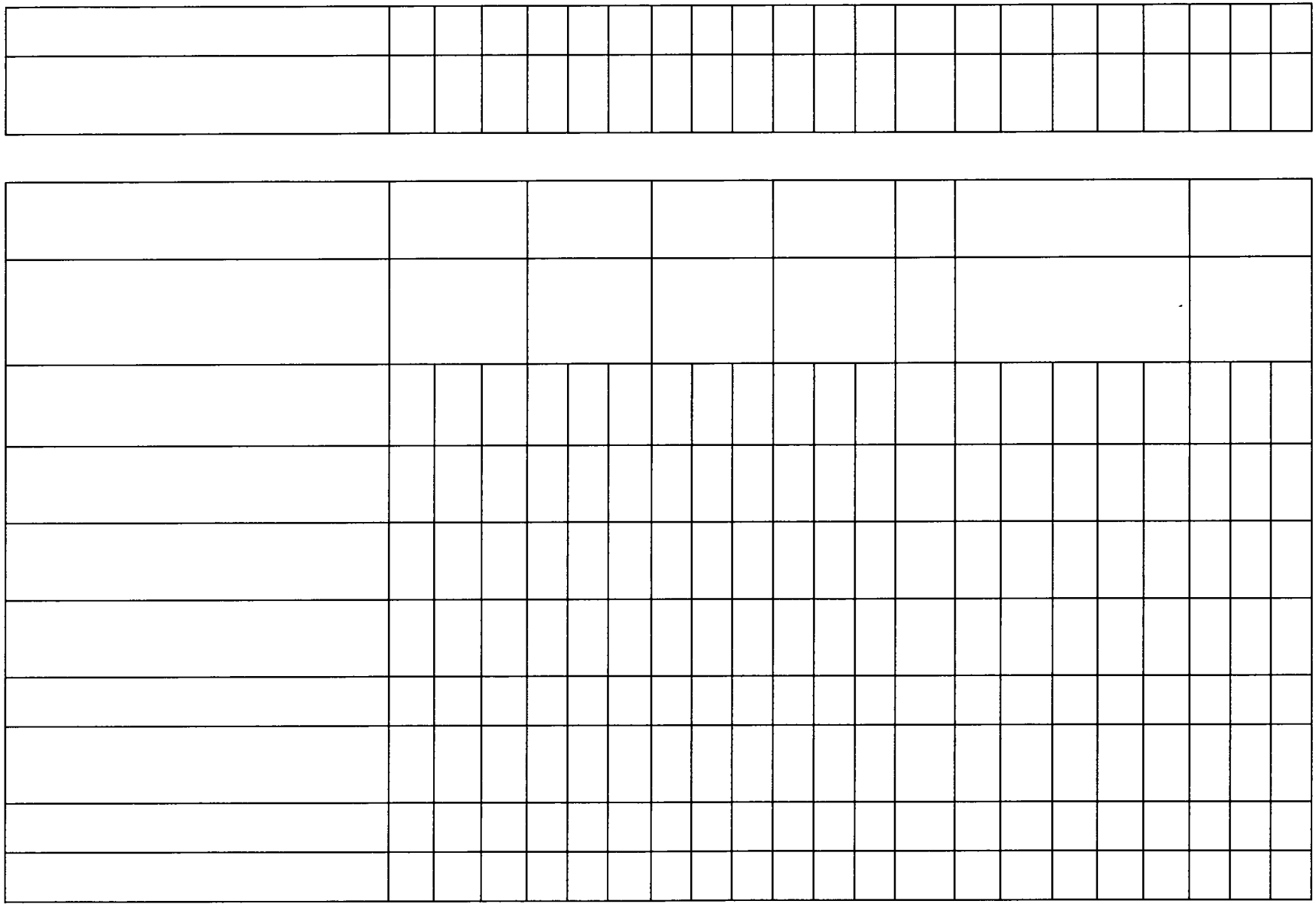


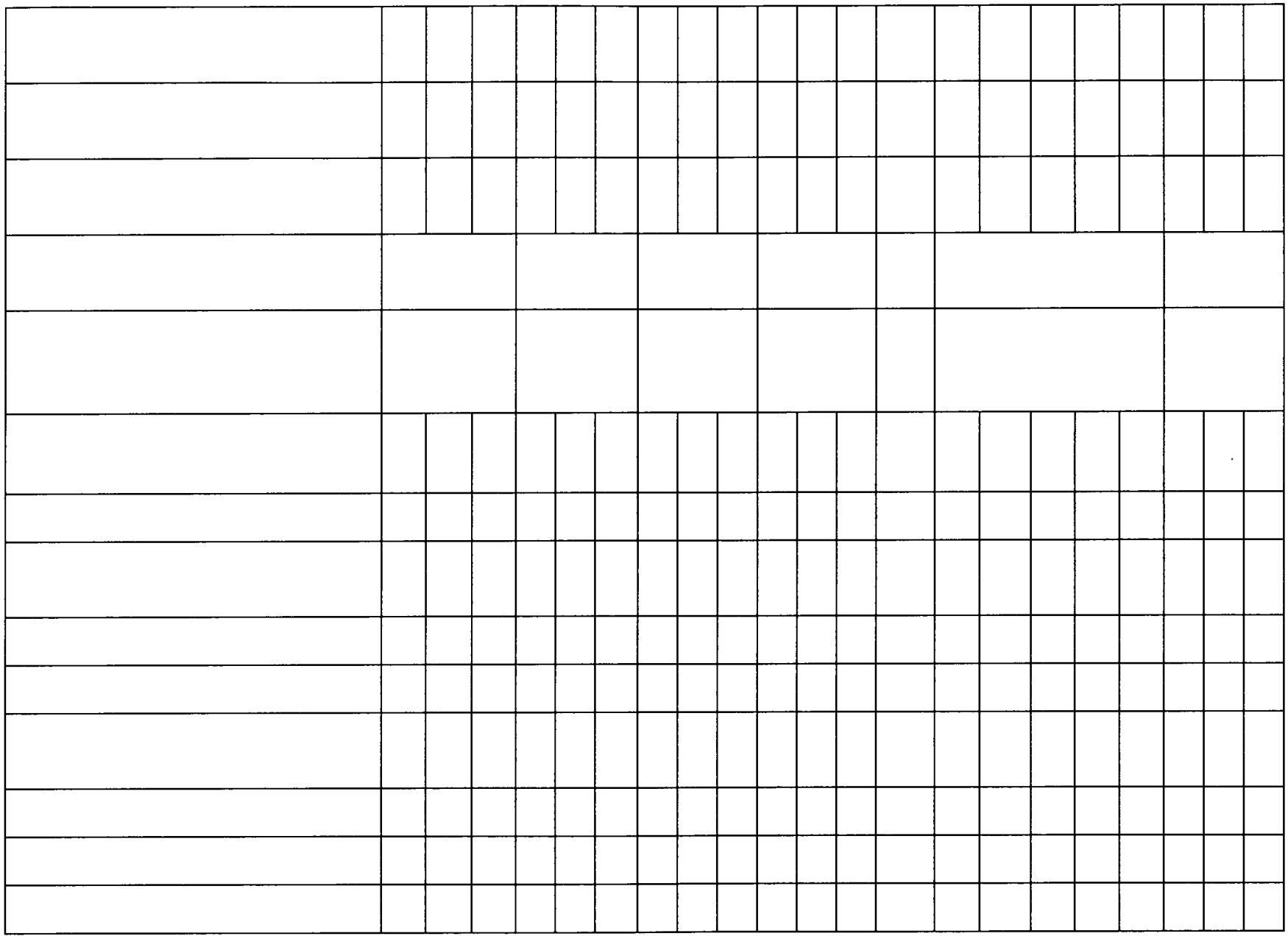


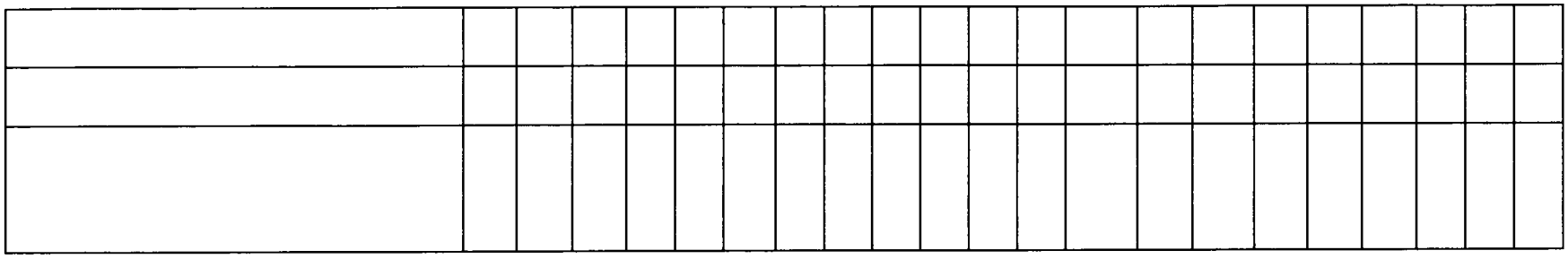


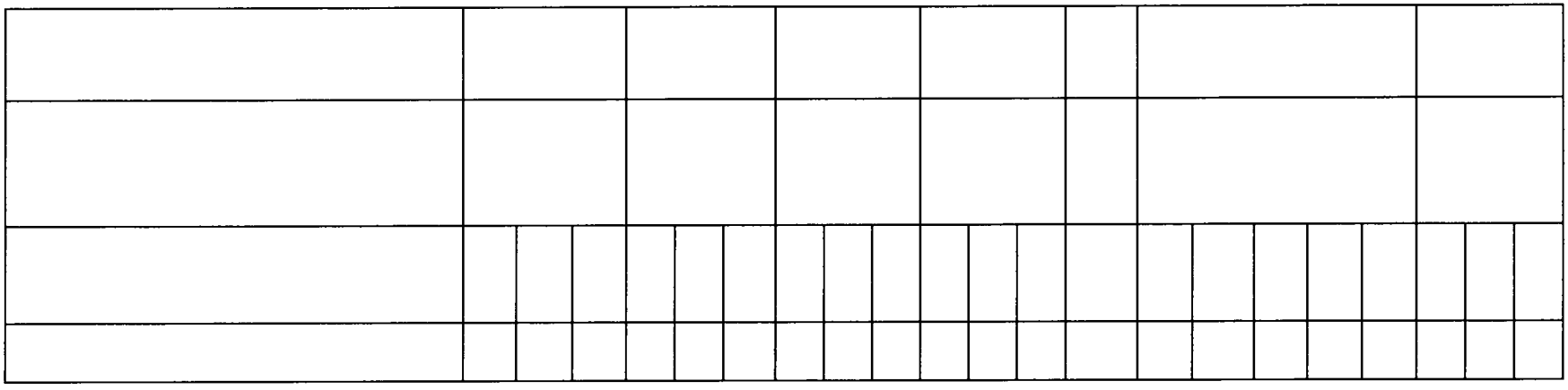












		]]

NEV80810

outbind://7-00000000718E93623C3BEE4AB0DC
07/24/07 07:34 AM

Zentis, Amber (GE Infra, Energy, Non-GE)

From: ENERGY GNF Communications (GE Infra, Energy)
Sent: Monday, July 23, 2007 5:15 PM
To: @ENERGY GNF-A Supervisors
Cc: Gurganious, Eddie (GE Infra, Aviation, US); Griffith, Chanel, HR (GE Infra, Aviation, US); Zentis, Amber (GE Infra, Energy, Non-GE)
Subject: ISQO Weekly Update FW30

ISQO Weekly FW30

Thad Leister's Update

Last week was tough for Fuels as we continue to battle issues with the Oracle ERP system. As w there are growing pains. I am asking for your continued patience as the IT team works diligently continue to elevate issues to your supervisors for attention.

Last week was also tough for production. Currently we are behind, to some degree, in all produc delivering a quality product at the required level to meet our customer commitments.

Integrity

One of the core Spirit and Letter policies for protecting GE's assets is the Conflict of Interest Polic illustrate that it is important that we continue to identify and address any potential conflicts of ir policy is the regular disclosure of any outside activities, financial interests or relationships that m a conflict of interest. For more information about the Conflict of Interest Policy, please visit the G http://integrity.ge.com.

In addition to complying with the law and GE policies, exercise your own good judgment in all pe your job. Additionally, you should:

- Avoid actions or relationships that might conflict or appear to conflict with your job respc
- Do not misuse GE resources, intellectual property, time or facilities (including office equip for personal gain.
- Before accepting any position as an officer or director of an outside business concern:
  - Consider the advantages and disadvantages to GE, including the appearance of p
  - Consider your responsibilities as a director as specified by laws and regulations. ("Additional responsibilities of leaders" section of the policy.
- Obtain the approval of your manager when accepting a board position with a not-for p business relationship with the entity or an expectation of financial or other support from
- Disclose your outside activities, financial interests or relationships that may present a pos of a conflict) to your manager as well as your business' legal counsel or finance manager such a situation arises as well as when asked to complete a "Conflict of Interest" questior
- If you know of a possible conflict of interest involving another employee or anyone else re manager, company legal counsel, GE auditor, GE ombudsperson or other GE compliance written or oral, and it may be anonymous.
- Follow the basic employee responsibilities common to all policies, which you can find onli

Safety

Again, I would like to thank the 2nd Quarter Wilmington Cup winning teams. Congratulations to: FCO Tubing-McCoy, Gas-Guzzling-Grimstead, Warehouse, DCP-C-Jackman-Jolin and Lab-Analy:

So how are we doing this year with work related injuries? Injuries are down 32% from this point l 26% as compared to last year's 42% through the middle of July 2006. Lets keep up the good wo Be aware when things "don't seem right" and pursue an answer why. Any and all injuries, regarc immediately.

Quality

Last week Western Zirc notified us that as a result of some non-routine testing they detected iro

	listed in our zirc-alloy tubing liner specifications. The results indicate some variation across the i material. As a result it is believed that this variation has always existed. Engineering has compl and has concluded that there is no adverse affect on quality as a result of this discovery. In fact range iron content material as a product. Per the ASTM sampling guidelines this material is with documenting the finding and evaluating any necessary changes to the supply chain for zirc. Southern Nuclear Company will be on site Wednesday and Thursday of this week for their Annu will tour FMO, FCO and SCO during their visit. Please answer questions and assist them as need
Output	**Fuel Fab:** Overall pellet yield for last week was 90.3%, a slight improvement over previous week Ceramics production was off last week due predominately to on boarding of new employees in t work hard toward equipment certification, particularly on the grinders. The GAD Wellman Furna week. This second operational furnace will improve throughput in the GAD Shop. I appreciate the extended hours the Ceramics employees have been putting in. We need to con production challenges of the third quarter as we transition to a fully functional three-shift rotatic **Bundle Assembly:** Bundle Assembly built 52 bundles last week, paced by (1) GAD rod availability regarding zirconium quality that was addressed by our supplier on Friday. The weekly productic the next ten weeks to meet liquidations targets. There are many challenges to face this quarter, so please stay focused on ISQO in all activities. **Powder Production:** DCP met the production schedule last week. We need to push 26K kg thro The team is working hard at meeting the goals and staying safe. **Tubing:** Last week, the tubing team completed 7750 tubes versus a schedule of 8300. This puts quarter schedule. The primary driver for the miss was staffing. A huge training effort is underwc being the resumption of a fully staffed three-shift operation. Everyone needs to use a good que transition to ensure we continue to deliver on safety and quality while ramping up our output. **Channels and Water Rods:** Channels completed 52 channels last week and is set up well for th with 108 single piece water rods and 236 GNF-J center tubes. The Water Rods team is on track Last week was a strong performance by both teams and they should continue to focus on meet customer requirements.
Shutdown	SCO will be shut down during FW31, July 30 through August 5, to complete equipment and facili

Got an Update? Email mailto:jody.farmer@gnf.com.