SOFTWARE QUALITY ASSURANCE PLAN

TEMPLATE

TM-SQA-01 V2.0

DECEMBER 16, 2003

Systems Engineering Process Office, Code 212
Space and Naval Warfare Systems Center San Diego
53560 Hull Street
San Diego, CA 92152-5001

PREFACE

This document is a template of a Software Quality Assurance (SQA) Plan using the guidelines provided in the Institute of Electrical and Electronics Engineers (IEEE) 730-1998, IEEE Standard for Software Quality Assurance Plans, and IEEE Std 730.1-1995, IEEE Guide for Software Quality Assurance Planning. This template should be supplemented with project-specific information to produce a SQA Plan that accurately describes the project's SQA organization, tasks, role, and responsibilities. The planning and documenting of SQA activities must agree and be consistent with the project's Software Development Plan (SDP) and any other project-planning document. Additionally, the SQA Plan must comply with Space and Naval Warfare (SPAWAR) Systems Center (SSC) San Diego SQA Policy, which provides management with appropriate visibility into the process being used by the software project and of the products being built.

This document supplements the SQA Process. Refer to Section 4, Create/Maintain SQA Plan, of the SQA Process for a description on the use of this template.

The SSC San Diego Systems Engineering Process Office (SEPO) assumes responsibility for this document and updates it as required to meet the needs of users within SSC San Diego CA. SEPO welcomes and solicits feedback from users of this document so that future revisions will reflect improvements, based on organizational experience and lessons learned.

Users of this document may report deficiencies or corrections using the Document Change Request (DCR) found on the next page or online through the SSC San Diego Process Asset Library (PAL) at http://sepo.spawar.navy.mil/. Updates are performed in accordance with the SEPO Configuration Management Procedure available on the SSC San Diego PAL.

DOCUMENT CHANGE REQUEST (DCR)

Document Title: **Software Quality Assurance Plan Template**	Tracking Number:
Name of Submitting Organization:	
Organization Contact:	Phone:
Mailing Address:	
Short Title:	Date:
Change Location: (use section #, figure #, table #, etc.)	
Proposed change:	
Rational for Change:	

Note: For the Systems Engineering Process Office (SEPO) to take appropriate action on a change request, please provide a clear description of the recommended change along with supporting rationale.

Send to: Commanding Officer, Space and Naval Warfare Systems Center, Code 212, 53560 Hull Street, San Diego, CA 92152-5001
Fax: (619) 553-6249
Email: sepo@spawar.navy.mil
Submit online: http://sepo.spawar.navy.mil/

DCR Form 9/2002

RECORD OF CHANGES

*__A__ - ADDED __M__ - MODIFIED __D__ - DELETED

VERSION NUMBER	DATE	NUMBER OF FIGURE, TABLE OR PARAGRAPH	**A* M D**	TITLE OR BRIEF DESCRIPTION	CHANGE REQUEST NUMBER
1.3	1/25/00	Entire Document		Updated Template to include checklists for general Software Engineering Process Verification (Appendix B) and focused CMM Key Process Area Verification (Appendix C)	
2.0	12/16/03	Entire Document		Revised Task definitions and reorganized them into a separate section; incorporated changes from DCR #SQAPT-003; removed SW-CMM Key Process Area validation process and checklists (to be documented as a separate document – addressing CMMI verification); added an "escalation procedure" to Section 7 for resolution of non-concurrence of SQA recommendations	SQAPT-003 SQA-0001

DOCUMENT CONVENTIONS

This document is a Software Quality Assurance (SQA) Plan template. As such, wording in this document should be supplemented with project-specific information to produce an SQA Plan that accurately describes the project SQA organization. Therefore, tailor (add, delete, change, or expand) the information provided in this document

Standard conventions are used within this document to direct the reader to specific sections of the text. These sections provide instructions and explanations and require users to substitute their own department-specific information for the generic information provided or to "fill in a blank."

[[text]] Global changes. Items that appear in regular text and are surrounded by double brackets represent changes that can be made globally throughout the document.

Italics Instructions and explanations. Items that appear in italics represent instructions to the user and are not to appear in the completed version of the document.

In some cases where information may already be found in another project document, like the Software Development Plan (SDP), refer to that document rather than duplicate the information in the project SQA Plan.

The template begins with the Project SQA cover sheet on the page after the next. Delete all pages prior to the Project SQA cover sheet in the final format of the project SQA Plan. Update the header page to reflect the document configuration identifier for the project SQA Plan.

This page intentionally left blank.

[[PROJECT NAME]]

SOFTWARE QUALITY ASSURANCE PLAN

[[CONFIGURATION CONTROL #]]

[[DOCUMENT DATE]]

[[Add your organization name here]]
Space and Naval Warfare Systems Center San Diego
53560 Hull Street
San Diego, CA 92152-5001

This page intentionally left blank.

[[PROJECT NAME]]

SOFTWARE QUALITY ASSURANCE PLAN

[[CONFIGURATION CONTROL #]]

[[DOCUMENT DATE]]

SQA Plan Approvals:

_____	_____
SQA Manager	**Date**
_____	_____
Project Manager	**Date**
_____	_____
Program Manager	**Date**

PREFACE

This document contains the Software Quality Assurance (SQA) Plan for the [[Project Name]]. The SQA activities described in this plan are consistent with the [[Project Name]] Software Development Plan *(or Project Management Plan)* and other project planning documents. This document has been tailored from the SQA Plan Template, TM-SQA-01, v2.0.

The [[Code/Project/Office]] assumes responsibility for this document and updates it, as required, to meet the needs of [[Project Name]]. Users of this document may report deficiencies or corrections using the Document Change Request found at the end of the document. Updates to this document will be performed, at least annually, in accordance with the [[Project Name Configuration Management Process]].

RECORD OF CHANGES

*__A__ - ADDED __M__ - MODIFIED __D__ - DELETED

VERSION NUMBER	DATE	NUMBER OF FIGURE, TABLE OR PARAGRAPH	**A* M D**	TITLE OR BRIEF DESCRIPTION	CHANGE REQUEST NUMBER

TABLE OF CONTENTS

Section **Page**

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

SECTION 1. PURPOSE

The purpose of this plan is to define the [[Project Name]] Software Quality Assurance (SQA) organization, SQA tasks and responsibilities; provide reference documents and guidelines to perform the SQA activities; provide the standards, practices and conventions used in carrying out SQA activities; and provide the tools, techniques, and methodologies to support SQA activities, and SQA reporting.

1.1 SCOPE

This plan establishes the SQA activities performed throughout the life cycle of the [[Project Name]].

This plan is written to follow the Space and Naval Warfare (SPAWAR) Systems Center (SSC) San Diego SQA policy, reference (a), for [[Project Name]]. Specifically, this SQA Plan will show that the SQA function is in place for this project. It will show that the SQA group has a reporting channel to senior management that is independent of the project manager, the project's software engineering group, and software related groups that includes Software Configuration Management (SCM), System and Software Test, and Logistics.

The goal of the SQA program is to verify that all software and documentation to be delivered meet all technical requirements. The SQA procedures defined herein shall be used to examine all deliverable software and documentation to determine compliance with technical and performance requirements.

Table 1-1 shows the software life cycle activities of the Configuration Items (CIs), as identified by the Institute of Electrical and Electronics Engineers (IEEE)/Electronic Industries Association (EIA) Standard 12207 Series, Software Life Cycle Process, reference (b), to which this SQA Plan applies.

In Table 1-1, add or delete activities that apply to the project's software lifecycle and as specified in the project's Software Development Plan (SDP) or other planning document.

1.2 IDENTIFICATION

Table 1-2 shows the CIs to which this plan applies.

If the project chooses to reference the list of CIs from another document, put a pointer here that shows where the project keeps its list of CIs.

Listed below is a brief description of each of the CIs developed and maintained by [[Project Name]].

a. [[**CI #1**]] - [[Include a brief description of the CI and its purpose]].

b. [[**CI #2**]] - [[Include a brief description of the CI and its purpose]].

c. [[**CI #3**]] - [[Include a brief description of the CI and its purpose]].

1.3 SYSTEM OVERVIEW

The [[System Name]] c*omplete the sentence by providing a description of the system and the intended use of the system.* The system includes [[enter the number of subsystems, e.g., 4]] subsystem(s) within the system. Figure 1-1 identifies the CIs within each subsystem and highlights those to which this SQA Plan applies.

TABLE 1-1. SOFTWARE LIFECYCLE ACTIVITIES

SOFTWARE LIFECYCLE ACTIVITY
Project Planning and Oversight
Software Development Environment
System Requirements Analysis
System Design
Software Requirements Analysis
Software Design
Software Implementation and Unit Testing
Unit Integration and Testing
CI Qualification Testing
CI/Hardware Configuration Item (HWCI) Integration and Testing
System Qualification Testing
Software Use Preparation
Software Transition Preparation
Life Cycle Maintenance

TABLE 1-2. CI NOMENCLATURE/IDENTIFICATION

NOMENCLATURE	ACRONYM	CI NUMBER
[[CI Name]]	[[Acronym]]	[[CI ID number]]
[[CI Name]]	[[Acronym]]	[[CI ID number]]
[[CI Name]]	[[Acronym]]	[[CI ID number]]

1.4 DOCUMENT OVERVIEW

This document identifies the organizations and procedures to be used to perform activities related to the [[Project Name]] SQA program as specified in IEEE Std 730-1998, IEEE Standard for Software Quality Assurance Plans, reference (c) and IEEE Std 730.1-1995, IEEE Guide for SQA Planning, reference (d).

Section 1 identifies the system to which this SQA Plan applies; provides an overview of the system and its software functions; summarizes the purpose and contents of the SQA Plan; and describes the relationship of the SQA Plan to other management plans and lists all documents referenced in this SQA Plan.

Section 2 describes each major element of the organization that influences the quality of the software.

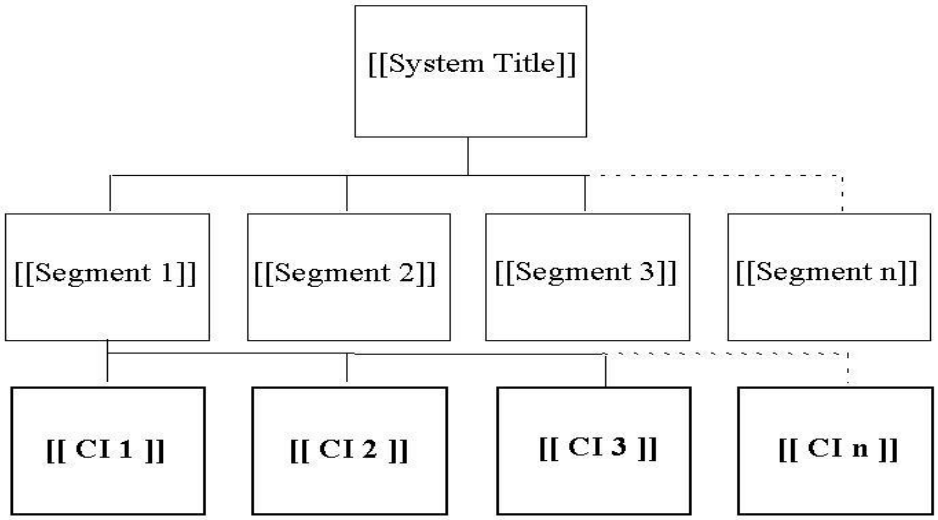

Figure 1-1. [[System Title]] Software

Section 3 describes the various SQA tasks

Section 4 lists the baseline documents produced and maintained by the project.

Section 5 identifies the standards, practices and conventions.

Section 6 describes SQA involvement in testing.

Section 7 describes problem reporting and corrective action.

Section 8 describes SQA tools, techniques, and methodologies.

Section 9 describes the configuration management tool used for code control.

Section 10 describes SQA evaluation of media control.

Section 11 describes control of supplier software.

Section 12 describes SQA procedures for record collection, maintenance, and retention.

Section 13 describes SQA training requirements.

Section 14 describes SQA review of the Risk Management process.

Appendix A provides a list of acronyms.

Appendix B provides checklists to be used or tailored for verifying compliance with general software engineering best practices.

1.5 RELATIONSHIP TO OTHER PLANS

SQA evaluation of the software development processes throughout the life cycle is based on the processes defined in the [[Project Name]] Software Development Plan (SDP), reference (e). Reference (e) and its implementation procedures establish the SQA evaluation criteria. The SQA Plan is implemented in conjunction with the [[Project Name]] Software Configuration Management Plan, reference (f).

1.6 REFERENCE DOCUMENTS

This section lists the documents referenced in this SQA Plan.

For the following, add or delete documents that were referenced in the SQA Plan.

a. SSC San Diego Software Quality Assurance Policy, Version 1.1, October 1997.

b. IEEE/EIA Standard 12207 Series - Standard for Information Technology – Software life cycle processes, March 1998.

c. IEEE-Std-730-1998, IEEE Standard for Software Quality Assurance Plans, June 1998.

d. IEEE-Std-730.1-1995, IEEE Guide for Software Quality Assurance Planning, December 1995.

e. [[Project Name]] Software Development Plan, [[Documentation Identification]], [[Document Date]].

f. [[Project Name]] Software Configuration Management Plan, [[Documentation Identification]], [[Document Date]].

g. Software Engineering Process Policy, SPAWARSYSCEN SAN DIEGO INST 5234.1, July 2000.

h. [[Program Title]] Program Plan, [[Documentation Identification]], [[Document Date]].

i. Software Quality Assurance Process, PR-SQA-02.

j. Software Quality Assurance Plan Template, TM-SQA-01.

k. A Description of the Space and Naval Warfare System Center San Diego Software Process Assets, PR-OPD-03.

l. MIL-STD-1521, Technical Reviews and Audits for Systems, Equipments, and Computer Software.

m. MIL-STD-973, Configuration Management. NOTE: This standard has been superceded by EIA-649, the Consensus Standard for Configuration Management, but the provisions of MIL-STD-973 are considered useful as a guide for developing Software Configuration Management activities.

n. Peer Review Process, PR-PR-02.

o. Military Standard (MIL-STD)-498, Software Development and Documentation, Data Item Descriptions (DIDs). NOTE: Although MIL-STD-498 has been superceded by IEEE Std 12207, the DIDs for MIL-STD-498 are still considered applicable for the support of developing software engineering procedures and supporting documentation.

p. IEEE Std 1045-1992, IEEE Standard for Software Productivity Metrics, September 1992.

q. IEEE Std 1061-1992, IEEE Standard for a Software Quality Metrics Methodology, December 1992.

r. IEEE Std 982.1-1988, IEEE Standard Dictionary of Measures to Produce Reliable Software, June 1988.

s. Std 982.2-1988, IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software, September 1988.

SECTION 2. MANAGEMENT

This section describes each major element of the organization that influences the quality of the software.

2.1 ORGANIZATION

Good software practice requires a measure of independence for the SQA group. This independence provides a key strength to SQA; that is, SQA has the freedom, if the quality of the product is being jeopardized, to report this possibility directly above the level of the project. While in practice this rarely occurs, for almost all problems are correctly addressed at the project level, the fact that the SQA group can go above the project level gives it the ability to keep many of these problems at the project level.

Figure 2-1 shows the SQA organization with relation to the project organization.

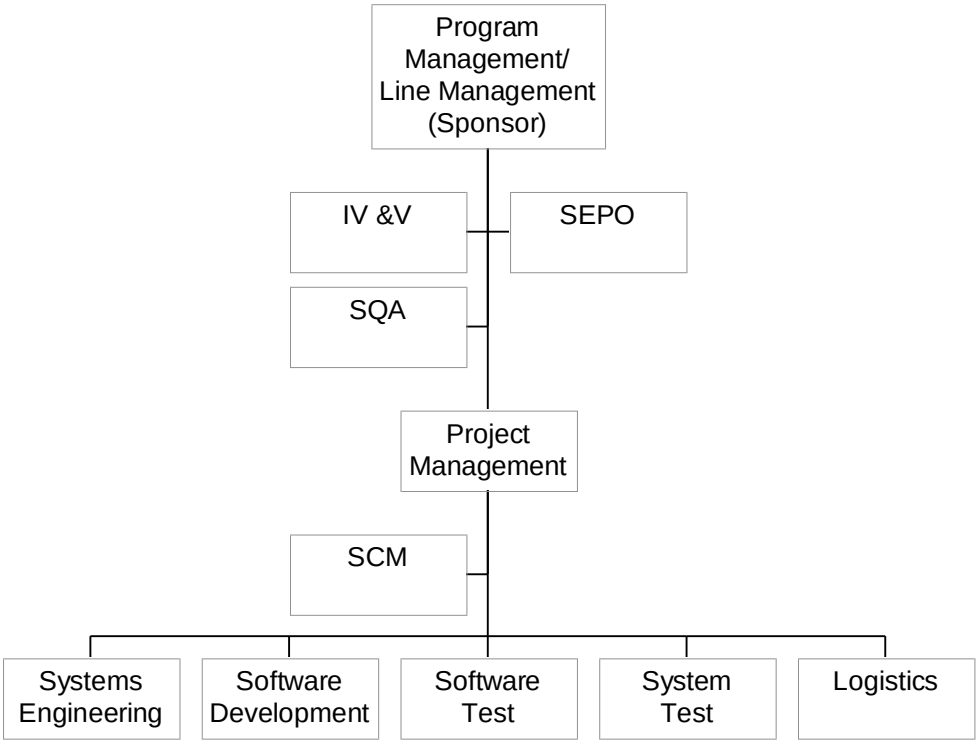

Figure 2-1. [[Project Name]] Organization

Replace Figure 2-1 with your project's organizational structure or reference the organizational chart's location. The project may wish to keep a single chart in a central location and reference all of its plans and procedures to that chart to facilitate maintaining the organization char. Provide a description of the functional responsibilities for each functional group in the organizational structure.

In describing the functional responsibilities, answer the questions listed below:

a. Who interacts with SQA?

b. Who has authority and delegates responsibilities of interacting functions?

c. What are the reporting relationships among the interacting elements identifying independence/dependence?

d. Who has product release authority?

e. Who approves the SQA Plan?

f. What are the reporting lines for escalating conflicts and the method by which conflicts are to be resolved among the elements?

In each case, add or delete the functional responsibilities that apply.

SQA is responsible for ensuring compliance with SQA requirements as delineated in this SQA Plan. The SQA organization assures the quality of deliverable software and its documentation, non-deliverable software, and the engineering processes used to produce software.

The following describes the functional groups that influence and control software quality.

a. Program Management/Line Management (Sponsor) is responsible for the following items:

1. Establishing a quality program by committing the project to implement the Software Engineering Process Policy, reference (g), and reference (a).
2. Reviewing and approving the [[Project Name]] SQA Plan.
3. Resolving and following-up on any quality issues raised by SQA.
4. Identifying an individual or group independent from the project to audit and report on the project's SQA function.
5. Identifying the quality factors to be implemented in the system and software.
6. *fill-in additional functional responsibilities.*

b. Project Management is responsible for:

1. Implementing the quality program in accordance with references (g) and (a).
2. Identifying the SQA activities to be performed by SQA.
3. Reviewing and approving the [[Project Name]] SQA Plan.
4. Identifying and funding an individual or group independent from the project to perform the SQA functions.
5. Resolving and following-up on any quality issues raised by SQA.
6. Identifying and ensuring the quality factors to be implemented in the system and software.
7. Identifying, developing and maintaining planning documents such as the Program Management Plan, reference (h), references (e) and (f), Test Plans, and this SQA Plan.
8. *fill-in additional functional responsibilities.*

c. System Engineering is responsible for:

1. Reviewing and commenting on the [[Project Name]] SQA Plan.
2. Implementing the quality program in accordance with this SQA Plan.
3. Resolving and following-up on any quality issues raised by SQA related to software engineering activities.
4. Identifying, implementing, and evaluating the quality factors to be implemented in the system (software and hardware).
5. Implementing the engineering practices, processes, and procedures as defined in reference (e) and other program/project planning documents.

6. *fill-in additional functional responsibilities.*

d. Software Design/Development is responsible for:

1. Reviewing and commenting on the [[Project Name]] SQA Plan.
2. Implementing the quality program in accordance with this SQA Plan.
3. Resolving and following-up on any quality issues raised by SQA related to software design and development.
4. Identifying, implementing, and evaluating the quality factors to be implemented in the software.
5. Implementing the software design/development practices, processes, and procedures as defined in reference (e) and other program/project planning documents.
6. *fill-in additional functional responsibilities.*

e. Software Test is responsible for:

1. Reviewing and commenting on the [[Project Name]] SQA Plan.
2. Implementing the quality program in accordance with this SQA Plan.
3. Resolving and following-up on any quality issues raised by SQA related to software test.
4. Verifying the quality factors are implemented in the system, specifically software.
5. Implementing the software test practices, processes, and procedures as defined in reference (e) and other program/project planning documents.
6. *fill-in additional functional responsibilities.*

f. System Test is responsible for:

1. Reviewing and commenting on the [[Project Name]] SQA Plan.
2. Implementing the quality program in accordance with this SQA Plan.
3. Resolving and following-up on any quality issues raised by SQA as related to system test.
4. Verifying the quality factors are implemented in the system (software and hardware).
5. Implementing the system test practices, processes, and procedures as defined in reference (e) and other program/project planning documents.
6. *fill-in additional functional responsibilities.*

g. Logistics is responsible for:

1. Reviewing and commenting on the [[Project Name]] SQA Plan.
2. Implementing the quality program in accordance with this SQA Plan.
3. *fill-in additional functional responsibilities.*

h. Software Configuration Management (SCM) is responsible for:

1. Reviewing and commenting on the [[Project Name]] SQA Plan.
2. Implementing the quality program in accordance with this SQA Plan.
3. Resolving and following-up on any quality issues raised by SQA related to SCM.

4. Ensuring the quality factors are implemented in the software related to SCM.
5. Implementing the SCM practices, processes, and procedures as defined in reference (e) and other program/project planning documents.
6. *fill-in additional functional responsibilities.*

i. Independent Verification and Validation (IV&V) is responsible for:

1. Reviewing and commenting on the [[Project Name]] SQA Plan.
2. Implementing the quality program in accordance with the [[Project Name]] SQA Plan.
3. Resolving and following-up on any quality issues raised by SQA.
4. Verifying the quality factors are implemented in the system (hardware and software).
5. Implementing the practices, processes, and procedures as defined for IV&V in reference (e) and other program/project planning documents.
6. *fill-in additional functional responsibilities.*

j. Systems Engineering Process Office (SEPO) is responsible for:

1. Keeping references (a) and (g) current.
2. Maintaining the SQA Process, reference (i) and the SQA Plan Template, reference (j).
3. Ensuring SQA training availability, either by vendor or SEPO.
4. Providing assistance in software process engineering and software process improvement.
5. Maintaining the Description of the Space and Naval Warfare System Center San Diego Software Process Assets, reference (k), that describes many artifacts used to support SQA verification activities.

2.2 RESOURCES

2.2.1 Facilities and Equipment

SQA will have access to the facilities and equipment as described in reference (e). SQA will have access to computer resources to perform SQA functions such as process and product evaluations and audits.

2.2.2 Personnel

The SQA effort for this project is *person-year effort or indicate the amount of effort if it is less than 100% - ensure the effort agrees with the project Work Breakdown Structure.*

Identify the qualification requirements of the SQA Manager

The SQA Manager will be familiar with and will be able to apply the standards and guidelines listed in Section 1.6. Additionally, the SQA Manager will be familiar with software quality, software development-related activities, and structured analysis, design, coding, and testing.

SECTION 3. SQA TASKS

Describe the portion of the software life cycle covered by this SQA Plan, the tasks to be performed with special emphasis on SQA activities, and relationship between these tasks and the planned major checkpoints. The sequence of the tasks should be indicated. Tailor this section to reflect those tasks being verified that relate to the project's current/projected activities.

The scheduling of SQA tasks is driven by the software development schedule. Therefore, an SQA task is performed in relationship to what software development activities are taking place. One or more SQA tasks can be performed concurrently until a task is completed. A task is considered completed when the required report e.g., SQA Reports, Process Audits Reports, etc. are satisfactorily completed or action items have been closed. The following tasks, requiring coordination and cooperation with the project team, shall be performed by SQA.

3.1 TASK: REVIEW SOFTWARE PRODUCTS

Reference (e) identifies all software products to be developed and evaluated, and includes the standards or guidelines to be followed. As required, SQA shall assist the project in identifying the standards or guidelines to be followed in developing the software product. Section 4 lists the software products to be evaluated by SQA and describes the review process to be followed.

3.2 TASK: EVALUATE SOFTWARE TOOLS

SQA shall conduct evaluations of tools, both existing and planned, used for software development and support. The tools are evaluated for adequacy by assessing whether they perform the functions for which the tools are intended and for applicability by assessing whether the tool capabilities are needed for the software development or support. Planned tools are evaluated for feasibility by assessing whether they can be developed with the techniques and computer resources available or by procurement. Section 7 provides the format for reporting the results of a software tool evaluation.

3.3 TASK: EVALUATE FACILITIES

SQA shall evaluate facilities, both existing and planned, for adequacy by assessing whether they provide the needed equipment and space used for software development and support. Section 7 provides the format for reporting the results of evaluating the project's facilities.

3.4 TASK: EVALUATE SOFTWARE PRODUCTS REVIEW PROCESS

This SQA task assures that quality review processes are in place for all software products, which may include representations of information other than traditional hard-copy documents, and that these products have undergone software product evaluation, testing, and corrective action as required by the standard.

SQA shall check that software products that are ready for review are reviewed, verify results are reported and issues or problems reported are resolved in accordance with reference (e).

The results of this task shall be documented using the Process Audit Form described in Section 7 and provided to project management. SQA recommendation for corrective action requires project management's disposition and will be processed in accordance with the guidance in Section 7.

3.5 TASK: EVALUATE PROJECT PLANNING, TRACKING AND OVERSIGHT PROCESSES

Project planning, tracking and oversight involves project management to develop and document plans for Software Development, CI and System Test, Software Installation, and Software Transition. Section 1.6 lists the program/project plans. For project documents to be developed, SQA will assist in identifying the appropriate guidelines, standards, or Data Item Description (DIDs), and will assist with the tailoring of those guidelines, standards, or DIDs to meet the project's needs.

SQA shall evaluate that the project conducts the relevant activities stated in the Program and Project plans. To verify that these activities are performed as planned, SQA will audit the processes that define the activity, and will use reference (e) or other planning document as the measure of whether those activities are being met.

SQA will use the audit checklists in Figures B-1 and B-2 as guides for conducting the evaluation.

The results of this task shall be documented using the Process Audit Form described in Section 7. Any recommended changes to those plans will require update and approval by project management in accordance with the configuration management procedure as described in the [[Project Name]] SCM Plan.

3.6 TASK: EVALUATE SYSTEM REQUIREMENTS ANALYSIS PROCESS

Requirements analysis establishes a common understanding of the customer's requirements between that customer and the software project team. An agreement with the customer on the requirements for the software project is established and maintained. This agreement is known as allocating system requirements to software and hardware. Section 4 lists the system requirements documents.

SQA activities are listed below:

a. Verify that the correct participants are involved in the system requirements analysis process to identify all user needs.

b. Verify that requirements are reviewed to determine if they are feasible to implement, clearly stated, and consistent.

c. Verify that changes to allocated requirements, work products and activities are identified, reviewed, and tracked to closure.

d. Verify that project personnel involved in the system requirements analysis process are trained in the necessary procedures and standards applicable to their area of responsibility to do the job correctly.

e. Verify that the commitments resulting from allocated requirements are negotiated and agreed upon by the affected groups.

f. Verify that commitments are documented, communicated, reviewed, and accepted.

g. Verify that allocated requirements identified as having potential problems are reviewed with the group responsible for analyzing system requirements and documents, and that necessary changes are made.

h. Verify that the prescribed processes for defining, documenting, and allocating requirements are followed and documented.

i. Confirm that a configuration management process is in place to control and manage the baseline.

j. Verify that requirements are documented, managed, controlled, and traced (preferably via a matrix).

k. Verify that the agreed upon requirements are addressed in the SDP.

SQA may use the audit checklist in Figure B-3 as a guide for conducting the evaluation.

The results of this task shall be documented using the Process Audit Form described in Section 7 and provided to project management. SQA recommendation for corrective action requires project management's disposition and will be processed in accordance with the guidance in Section 7.

3.7 TASK: EVALUATE SYSTEM DESIGN PROCESS

The purpose of the system design process is to develop decisions about the system's behavioral design and other decisions affecting the selection and design of system components. System architectural design is organizing a system into subsystems, organizing a subsystem into Hardware Configuration Items (HWCIs), CIs, and manual operations, or other variations as appropriate. Section 4 lists the system design documents.

SQA activities are listed below:

a. Verify that system design documents and the traceability matrix are prepared and kept current and consistent.

b. Verify that relevant documents are updated and based on approved requirements changes.

c. Verify that design walkthroughs (peer reviews) evaluate compliance of the design to the requirements, identify defects in the design, and evaluate and report design alternatives.

d. Participate in a sampled set of design walkthroughs and verify all walkthroughs are conducted.

e. Identify defects, verify resolution for previous identified defects, and verify change control integrity.

f. Selectively review and audit the content of system design documents.

g. Identify lack of compliance with standards and determine corrective actions.

h. Determine whether the requirements and accompanying design and tools conform to standards, and whether waivers are needed prior to continuing software development.

i. Review demonstration prototypes for compliance with requirements and standards.

j. Verify that the demonstration conforms to standards and procedures.

k. Review the status of design milestones.

SQA may use the audit checklist in Figure B-4 as a guide for conducting the evaluation.

The results of this task shall be documented using the Process Audit Form described in Section 7 and provided to project management. SQA recommendation for corrective action requires project management's disposition and will be processed in accordance with the guidance in Section 7.

3.8 TASK: EVALUATE SOFTWARE REQUIREMENTS ANALYSIS PROCESS

The purpose of software requirements analysis is to formulate, document and manage the software requirements baseline; respond to requests for clarification, correction or change; analyze impacts; revise the software requirements specification; and manage the software requirements analysis and change process. Section 4 lists the software requirements documents.

SQA activities are listed below:

a. Verify that the software requirements analysis process and associated requirements reviews are conducted in accordance with the standards and procedures established by the project and as described in the SDP.

b. Verify that action items resulting from reviews of the software requirements analysis are resolved in accordance with these standards and procedures.

SQA may use the audit checklist in Figure B-5 as a guide for conducting the evaluation.

The results of this task shall be documented using the Process Audit Form described in Section 7 and provided to project management. SQA recommendation for corrective action requires project management's disposition and will be processed in accordance with the guidance in Section 7.

3.9 TASK: EVALUATE SOFTWARE DESIGN PROCESS

Preliminary design activity determines the overall structure of the software to be built. Based on the requirements identified in the previous phase, the software is partitioned into modules, and the function(s) of each module and relationships among these modules are defined.

A goal of detailed design is to define logically how the software will satisfy the allocated requirements. The level of detail of this design must be such that the coding of the computer program can be accomplished by someone other than the original designer. Section 4 lists the software design documents.

SQA activities are listed below:

a. Verify that the software design process and associated design reviews are conducted in accordance with standards and procedures established by the project and as described in the SDP.

b. Verify that action items resulting from reviews of the design are resolved in accordance with these standards and procedures.

c. Evaluate the method used for tracking and documenting the development of a software unit to determine the method's utility as a management tool for assessing software unit development progress. Example criteria to be applied for the evaluation are the inclusion of schedule information, results of audits, and an indication of internal review and approval of its constituent parts.

d. Verify that the method, such as the Software Development File (SDF) or Unit Development folder (UDF), used for tracking and documenting the development of a software unit is implemented and is kept current.

SQA may use the audit checklist in Figure B-6 as a guide for conducting the evaluation.

The results of this task shall be documented using the Process Audit Form described in Section 7 and provided to project management. SQA recommendation for corrective action requires project management's disposition and will be processed in accordance with the guidance in Section 7.

3.10 TASK: EVALUATE SOFTWARE IMPLEMENTATION AND UNIT TESTING PROCESS

Software implementation or coding is the point in the software development cycle at which the design is finally implemented. The process includes unit testing of the software code.

SQA activities are listed below:

a. Verify that the coding process, associated code reviews, and software unit testing are conducted in conformance with the standards and procedures established by the project and as described in the SDP.

b. Verify that action items resulting from reviews of the code are resolved in accordance with these standards and procedures.

c. Verify that the SDF used for tracking and documenting the development of a software unit is implemented and is kept current.

SQA may use the audit checklist in Figure B-7 as a guide for conducting the evaluation.

The results of this task shall be documented using the Process Audit Form described in Section 7 and provided to project management. SQA recommendation for corrective action requires project management's disposition and will be processed in accordance with the guidance in Section 7.

3.11 TASK: EVALUATE UNIT INTEGRATION AND TESTING, CI QUALIFICATION TESTING, CI/HWCI INTEGRATION AND TESTING, AND SYSTEM QUALIFICATION TESTING PROCESSES

Software integration and test activities combine individually developed components together in the developing environment to verify that they work together to complete the software and system functionality. For joint hardware and software development, integration requires close synchronization of hardware and software to meet designated integration and test milestones.

In the integration and test phase of the development lifecycle, the testing focus shifts from an individual component correctness to the proper operation of interfaces between components, the flow of information through the system, and the satisfaction of system requirements.

SQA activities are listed below:

a. Verify that software test activities are identified, test environments have been defined, and guidelines for testing have been designed. SQA will verify the software integration process, software integration testing activities and the software performance testing activities are being performed in accordance with the SDP, the software design, the plan for software testing, and established software standards and procedures.

b. Verify any transfer of control of code to personnel performing software integration testing or software performance testing is being accomplished in accordance with established software standards and procedures.

c. Verify that as many of the software integration tests as necessary and all of the software performance tests are witnessed to verify that the approved test procedures are being followed, that accurate records of test results are being kept, that all discrepancies discovered during the tests are being properly reported, that test results are being analyzed, and the associated test reports are completed.

d. Verify that discrepancies discovered during software integration and performance tests are identified, analyzed, documented, and corrected; software unit tests, and software integration tests are re-executed as necessary to validate corrections made to the code; and the software unit's design, code, and test is updated based on the results of software integration testing, software performance testing, and corrective action process.

e. Verify the software performance tests produce results that will permit determination of performance parameters of the software.

f. Verify that the responsibility for testing and for reporting on results has been assigned to a specific organizational element.

g. Verify that procedures are established for monitoring informal testing.

h. Review the Software Test Plan and Software Test Descriptions for compliance with requirements and standards.

i. Verify that the software is tested.

j. Monitor test activities, witness tests, and certify test results.

k. Verify that requirements have been established for the certification or calibration of all support software or hardware used during tests.

SQA may use the audit checklists in Figures B-8 and B-9 as guides for conducting these evaluations.

The results of this task shall be documented using the Process Audit Form described in Section 7 and provided to project management. SQA recommendation for corrective action requires project management's disposition and will be processed in accordance with the guidance in Section 7.

3.12 TASK: EVALUATE END-ITEM DELIVERY PROCESS

This activity is applicable for those projects providing a one-time delivery of a product and may also be interpreted as required deliveries for a specified time period or time frame.

SQA shall evaluate the activities in preparation for end-item delivery to verify that program or project requirement, if any, for functional and physical audits of the end-item products are being satisfied. In some cases, the SQA organization should be allowed to prohibit delivery of certain items, such as documentation, code, or a system, if the project fails to meet contractual requirements or standards.

SQA may use the audit checklist in Figure B-10 as a guide for conducting this evaluation.

The results of this task shall be documented using the Process Audit Form described in Section 7 and provided to project management. SQA recommendation for corrective action requires project management's disposition and will be processed in accordance with the guidance in Section 7.

3.13 TASK: EVALUATE THE CORRECTIVE ACTION PROCESS

The corrective action process describes the steps for (1) problem identification and correction occurring during software development to verify early detection of actual or potential problems, (2) reporting of the problem to the proper authority, (3) analysis of the problem to propose corrective measures, (4) timely and complete corrective action, and (5) the recording and follow-up of each problem's status. Problems in this context include documentation errors, software errors, and noncompliance with standards and procedures.

SQA activities are listed below:

a. Periodically review the corrective action process and their results against the SCM Plan to assess the effectiveness of the corrective action process.

b. Perform periodic analysis of all reported problems to identify trends that may disclose generic problem areas. These analyses shall include the study of the causes, magnitude of impact, frequency of occurrence, and preventive measures.

SQA may use the audit checklist in Figure B-11 as a guide for conducting this evaluation.

The results of this task shall be documented using the Process Audit Form described in Section 7 and provided to project management. SQA recommendation for corrective action requires project management's disposition and will be processed in accordance with the guidance in Section 7.

3.14 TASK: MEDIA CERTIFICATION

SQA shall verify that SCM certifies that the media containing the source code, and the media containing the object code which are delivered to the procuring agency, correspond to one another. SQA shall verify also that the software version represented by this media matches that on which software performance testing was performed, or correctly represents an authorized update of the code, as applicable.

SQA may use the audit checklist in Figure B-12 as a guide for conducting this evaluation.

SQA reports, together with the corrective action records, software test reports, and software product evaluation records can constitute the required evidence for certification.

The results of this task shall be documented using the Process Audit Form described in Section 7 and provided to project management. SQA recommendation for corrective action requires project management's disposition and will be processed in accordance with the guidance in Section 7.

3.15 TASK: NON-DELIVERABLE SOFTWARE CERTIFICATION

The project may use non-deliverable software in the development of deliverable software as long as the operation and support of the deliverable software after delivery to the acquirer do not depend on the non-deliverable software or provision is made to verify that the acquirer has or can obtain the same software. SQA shall certify that the use of non-deliverable software meets the above criteria, that is, deliverable software is not dependent on non-deliverable software to execute, or verify that the acquirer can obtain the same software. SQA shall verify that all non-deliverable software used on the project performs its intended functions.

SQA may use the audit checklist in Figure B-13 as a guide for conducting this evaluation.

SQA reports, together with the corrective action records, software test reports, and software product evaluation records can constitute the required evidence for certification.

The results of this task shall be documented using the Process Audit Form described in Section 7 and provided to project management. SQA recommendation for corrective action requires project management's disposition and will be processed in accordance with the guidance in Section 7.

3.16 TASK: EVALUATE STORAGE AND HANDLING PROCESS

SQA shall verify that there is an established plan, methodology, or set of procedures for storage and handling of the media. SQA shall evaluate the storage of the software product and documentation to verify that storage areas for paper products or media are free from adverse environmental effects such as high humidity, magnetic forces, and dust.

SQA may use the audit checklist in Figure B-14 as a guide for conducting this evaluation.

The results of this task shall be documented using the Process Audit Form described in Section 7 and provided to project management. SQA recommendation for corrective action requires project management's disposition and will be processed in accordance with the guidance in Section 7.

3.17 TASK: EVALUATE SUBCONTRACTOR CONTROL

SQA shall be responsible for ensuring that the quality of all software products from subcontractors conforms to the contract requirements and that the subcontractor's SCM plan and procedures are being followed.

SQA may use the audit checklist in Figure B-15 as a guide for conducting this evaluation.

SQA reports, together with the corrective action records, and software product evaluation records shall be provided to project management for corrective action by the subcontractor as required.

3.18 TASK: EVALUATE DEVIATIONS AND WAIVERS

SQA shall assist program or project management, with requests for deviations and waivers, if required, and verify that the deviation or waiver request is processed in accordance with the project's SCM Plan and approved by the approving agency.

SQA may use the audit checklist in Figure B-16 as a guide for conducting this evaluation.

3.19 TASK: EVALUATE CONFIGURATION MANAGEMENT PROCESS

CM is the discipline that applies technical and administrative direction and surveillance to (1) identify and document the functional and physical characteristics of CIs, (2) control the changes to CIs and their related documentation, (3) record and report information needed to manage CIs effectively, including the status of proposed changes and the implementation status of approved changes, and (4) audit the CIs to verify conformance to specifications, interface control documents, and other contract requirements.

SQA activities are listed below:

a. Verify that configuration identification of documents, code, and computer data has established standards for titling, naming, and describing change status.

b. Verify that baseline management of changes to the developmental baseline (including documents, code and computer data) are identified, reviewed, implemented, and incorporated in accordance with established procedures.

c. Verify configuration control of changes to baseline documents and software are being managed in accordance with SCM requirements as stated in the SCM Plan.

d. Verify configuration status accounting reports are prepared at established times, are prepared in accordance with established procedures, and report the status of items that are significant with respect to the management of the configuration of the software product and documentation.

e. Verify that the personnel assigned to participate in the configuration audits comply with the SCM Plan.

f. Verify for document control that only approved, up-to-date documentation is provided for use by project personnel, and that the document distribution process results in receipt of correct documentation.

g. Verify that the program support library is the single place of storage for the baseline version of all software. Verify that the identification of all software includes the software name and a unique version identifier. The evaluation shall also determine that control of access to software products is being properly exercised and that unauthorized changes to master files cannot occur.

SQA may use the audit checklist in Figure B-17 as a guide for conducting this evaluation.

The results of this task shall be documented using the Process Audit Form described in Section 7 and provided to project management. SQA recommendation for corrective action requires project management's disposition and will be processed in accordance with the guidance in Section 7.

3.20 TASK: EVALUATE SOFTWARE DEVELOPMENT LIBRARY CONTROL PROCESS

The SDL functions as the main control point for SCM. A SDL contains all units of code developed for evolving project CIs, as well as carefully identified listings, patches, errata, CI and system magnetic tapes and disk packs, and job control streams for operating or building software systems. The SDL also contains previous versions of the operational software system in the form of magnetic tapes or disk packs.

SQA activities are listed below:

a. Verify the establishment of the SDL and procedures to govern its operation.

b. Verify that documentation and computer program materials are approved and placed under library control.

c. Verify the establishment of formal release procedures for SCM approved documentation and software versions.

d. Verify that library controls prevent unauthorized changes to the controlled software and verify the incorporation of all approved changes.

SQA may use the audit checklist in Figure B-18 as a guide for conducting this evaluation.

The results of this task shall be documented using the Process Audit Form described in Section 7 and provided to project management. SQA recommendation for corrective action requires project management's disposition and will be processed in accordance with the guidance in Section 7.

3.21 TASK: EVALUATE NON-DEVELOPMENTAL SOFTWARE

Non-Developmental Software (NDS) is software that is provided by the contractor, the Government, or a third party. NDS may be referred to as reusable software, Government-furnished software, or commercially available software depending on it source. SQA shall verify that non-developmental software performs its intended functions.

SQA may use the audit checklist in Figure B-19 as a guide for conducting this evaluation.

SQA reports, together with the corrective action records, software test reports, and software product evaluation records can constitute the required evidence for certifying the software performs its intended functions.

3.22 TASK: VERIFY PROJECT REVIEWS AND AUDITS

Define the technical and managerial reviews and audits to be conducted. State how the reviews and audits are to be accomplished. State what further actions are required and how they are to be implemented and verified. The type and scope of technical reviews depends heavily on the size, scope, risk and criticality of the software project. The reviews and audits identified here should be the same as specified in reference (e).

Table 3-1 identifies the required reviews and audits for the system and software development phases.

3.22.1 Task: Verify Technical Reviews

A primary component of engineering quality into software is the conduct of technical reviews of software products, both deliverable and non-deliverable. Participants of a technical review shall include persons with technical knowledge of the software products to be reviewed. The purpose of the technical review will be to focus on in-progress and final software products rather than the materials generated especially for the review. SQA will assure that technical reviews are accomplished and will selectively attend them in accordance with approved sampling techniques. The guidelines of MIL-STD-1521B, Technical Reviews and Audits for Systems, Equipments, and Computer Software, reference (l), may be used for conducting a technical review. A summary of each kind of technical review appears below:

a. System Requirements Review (SRR) - the objective is to ascertain the adequacy of the developer's efforts in defining system requirements.

b. System Design Review (SDR) - the objective is to evaluate the optimization, correlation, completeness, and risks associated with the allocated technical requirements. Also included is a summary review of the system engineering process that produced the allocated technical requirements and of the engineering planning for the next phase of effort.

c. Software Specification Review (SSR) - the objective is to review the finalized CI requirements and operational concept. A successful SSR shows that the Software Requirements Specification (SRS), Interface Requirements Specification (IRS), and Operational Concept Document (OCD) form a satisfactory basis for proceeding into preliminary software design.

d. Software Preliminary Design Review (PDR) - the objective is to evaluate the progress, consistency, and technical adequacy of the selected top-level design and test approach, compatibility between software requirements and preliminary design, and the preliminary version of the operation and support documents.

e. Software Critical Design Review (CDR) - the objective is to determine acceptability of the detailed design, performance, and test characteristics of the design solution, and on the adequacy of the operation and support documents.

f. Software Test Readiness Review (TRR) - the objective is to determine whether the software test procedures are complete and to assure that the developer is prepared for formal CSCI/SU testing.

g. Formal Qualification Review (FQR) - the test, inspection, or analytical process by which a group of configuration items comprising the system are verified to have met specific program or project management performance requirements.

TABLE 3-1. REVIEWS AND AUDITS

SYSTEM AND SOFTWARE DEVELOPMENT PHASE	SOFTWARE PRODUCTS	REQUIRED AUDITS AND REVIEWS
System Requirements	(1) System/Subsystem Specification (SSS), IRS, OCD (2) SDP, SCM Plan, SQA Plan	(1) System Requirements Review (SRR) (2) Process Audits (3) Management Review (4) Peer Review

System Design	(1) System/Subsystem Design Description (SSDD), Interface Design Description (IDD)	(1) System Design Review (SDR) (2) Process Audits (3) Management Review (4) Peer Review
Software Requirements	(1) SRS, IRS	(1) Software Specification Review (SSR) (2) Process Audits (3) Management Review (4) Peer Review
Software Design	(1) Software Design Document (SDD), Database Design Description (DBDD), IDD	(1a) Software Preliminary Design Review (PDR) (1b) Software Critical Design Review (CDR) (2) Process Audits (3) Managerial Review (4) Peer Review
Software Implementation	Software products	(1) Process Audits (2) Management Review (3) Peer Review
Test	(1) Test Documentation	(1a) Software Test Readiness Review (TRR) (1b) Formal Qualification Review (FQR) (2) Process Audits (3) Managerial Review (4) Functional Configuration Audit (5) Peer Review
Software Release	(1) Software Version Description (SVD), User documentation	(1) Production Readiness Review (PRR) (2) Process Audits (3) Management Review (4) Physical Configuration Audit (5) Peer Review

Note: Peer review is discussed in Section 4.

h. Production Readiness Review (PRR) - the objective is to determine the status of completion of the specific actions that must be satisfactorily accomplished prior to executing a production go-ahead decision.

Technical reviews will be conducted to review evolving software products, demonstrate proposed technical solutions, and provide insight and obtain feedback on the technical effort. The outcome of a technical review is listed below:

a. Identify and resolve technical issues.

b. Review project status, specifically surface near- and long-term risk regarding technical, costs, and schedule issues.

c. Arrive at agreed-upon mitigation strategies for identified risks, within the authority of those present.

d. Identify risks and issues to be raised at joint management reviews.

e. Verify on-going communications between acquirer and developer technical personnel.

The entrance criteria for a technical review will require that an item to be reviewed is distributed to the group prior to the review meeting. Additionally, a recorder will be assigned to record any issues requiring resolution stating action item assignee and due date, and decisions made within the authority of the technical review participants.

Various measurements are collected as part of technical reviews to help determine the effectiveness of the review process itself as well as the process steps that are used to produce the item being reviewed. These measurements, reported to the project manager, will include the amount of time spent by each person involved in the review, including preparation for the review.

3.22.2 Task: Verify Management Reviews

SQA periodic management review of software project status, progress, problems, and risk will provide an independent assessment of project activities. SQA will provide the following information to management:

a. Compliance - Identification of the level of compliance of the project with established organizational and project processes.

b. Problem areas - identification of potential or actual project problem areas based on analysis of technical review results.

c. Risks - identification of risks based on participation and evaluation of project progress and trouble areas.

Because the SQA function is integral to the success of the project, SQA will freely communicate its results to senior management, project management and the project team. The method for reporting compliance, problem areas, and risks will be communicated in a documented report or memorandum. Compliance, problem areas, and risks will be followed-up and tracked to closure.

3.22.3 Task: Conduct Process Audits

Software development processes are audited according to the tasks specified in this Section and performed in accordance with the software development schedule specified in the SDP.

3.22.4 Task: Conduct Configuration Audits

3.22.4.1 Functional Configuration Audit. The Functional Configuration Audit (FCA) is held prior to software delivery to compare the software as built (including its executable forms and available documentation) with the software requirements as stated in the baseline SRS. The purpose is to ensure that the code addressed all, and only, the documented requirements and functional capabilities stated in the SRS. MIL-STD-973, Configuration Management, reference (m), provides the guidelines for conducting an FCA. SQA will participate as a member of the FCA team with other FCA team members to be assigned by the project manager. SQA will assist in the preparation of the FCA findings. Any follow-up to the reported FCA finding will be monitored and tracked to closure.

3.22.4.2 Physical Configuration Audit. The Physical Configuration Audit (PCA) is held to verify that the software and its documentation are internally consistent and are ready for delivery. The purpose is to assure that the documentation to be delivered is consistent and correctly describes the code. Reference (m) provides the guidelines for conducting a PCA. SQA will participate as a member of the PCA team

with other PCA team members to be assigned by the project manager. SQA will assist in the preparation of the PCA findings. Any follow-up to the reported PCA finding will be monitored and tracked to closure.

3.23 TASK: VERIFY SOFTWARE QUALITY ASSURANCE

The Project Manager requests periodic independent assessments of project SQA. These assessments will be conducted at least annually. The auditor, who must be independent of the assessed SQA group, will review SQA audits conducted on the project, including documented findings and corrective actions, and will consult with project personnel to ensure that SQA activities have been accomplished, and that corrective actions have been implemented or resolved. The auditor will report findings of the independent assessment to the Project and, where appropriate, Program Manager.

Independent assessments may be requested of higher-level SQA personnel (where available, Department-level SQA personnel) or from SEPO.

3.24 RESPONSIBILITIES

This paragraph should identify the specific organizational elements responsible for each task.

The ultimate responsibility for the quality of the [[Project Name]] software and associated documentation produced by [[SSC San Diego or Agency Name]] rests with the [[Project Name]] Software Project Manager. The SQA Manager shall implement the SQA procedures defined in this plan.

SQA derives its authority from the Project Manager through the [[SSC San Diego Branch/Division/Department or Agency Name]] Manager. SQA shall monitor project staff activities and review products for compliance to applicable standards, procedures, and reference (e). The results of SQA monitoring and analysis along with SQA recommendations for corrective action shall be reported to the Software Project Manager, and, as required, to the [[SSC San Diego Branch/Division/Department or Agency Name]] Manager. All documents and software approved by the Software Project Manager for release to [[user activity]] shall have been reviewed and approved by SQA. Table 3-2 is a responsibility matrix for the tasks identified in this Section.

TABLE 3-2. RESPONSIBILITY MATRIX

SQA Plan	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Develop/Document SQA Plan	X		X						
Review SQA Plan	X	X	X	X	X	X	X	X	X
Approve SQA Plan	X	X	X						

Review Software Products	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Review products	X	X	X	X	X	X	X	X	X

Rework by author	Applies as applicable								
Approve product		X	X						

Evaluate Software Tools	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Evaluate tool	X			X		X	X		
Resolve Audit Findings		X	X						

Evaluate Software Facilities	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Evaluate facilities	X					X	X		
Resolve Audit Findings		X	X						

Proj Planning, Tracking & Oversight (PPT&O) Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Develop/Document SDP and other project plans (Test Plan, Training Plan, Computer Resource Life Cycle Management Plan (CRLCMP))		X	X						
Review plans	X	X	X	X	X	X	X	X	X
Approve plans		X	X						
Evaluate PPT&O Process	X								
Resolve Audit Findings		X	X						

System Requirements Analysis Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**

Develop/document Sys Rqmts					X				X
CM Sys Rqmts				X					
Review Sys Rqmts	X	X	X		X	X	X	X	X
Approve Sys Rqmts		X	X						
Evaluate/report Sys Rqmts Analysis Process	X								
Resolve Audit Findings		X	X						

System Design Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Develop/document Sys Design					X				
CM Sys Design				X					
Review Sys Design	X	X	X		X	X	X	X	X
Approve Sys Design		X	X						
Evaluate/report Sys Design Process	X								
Resolve Audit Findings		X	X						

Software Requirements Analysis Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Develop/document SW Rqmts						X	X		
CM SW Rqmts				X					
Review SW Design	X	X	X		X	X	X	X	X
Approve SW Rqmts		X	X						
Maintain SDL and SDFs				X	X	X			
Evaluate/report SW Rqmts Analysis Process	X								

Resolve Audit Findings		X	X						

Software Design Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Develop/document SW Design						X	X		
CM SW Design				X					
Review SW Design	X	X	X		X	X	X	X	X
Approve SW Design		X	X						
Maintain SDL and SDFs				X		X			
Evaluate/report SW Design Process	X								
Resolve Audit Findings		X	X						

Software Implementation & Unit Testing Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Develop/fix code						X			
CM code				X					
Code review	X					X	X		
Unit Test						X	X		
Maintain SDL and SDFs				X		X	X		
Maintain STR process				X					
Evaluate/report SW Implementation and Unit Testing Process	X								
Resolve Audit Findings		X	X						

Unit Integration and Testing Process	SQA Mgr	Prog Mgr	Proj Mgr	SCM	Sys Eng	SW Dev	SW Test	Sys Test	Logi stics
Integrate SW						X			
Test Integrated SW							X	X	
Fix errors						X			
Maintain SDL and SDFs				X		X	X		
Maintain STR process				X					
Evaluate/report Unit Integration and Testing Process	X								
Resolve Audit Findings		X	X						

CI Qualification Testing Process	SQA Mgr	Prog Mgr	Proj Mgr	SCM	Sys Eng	SW Dev	SW Test	Sys Test	Logi stics
Performance Test							X	X	
Fix errors						X			
Maintain SDL and SDFs				X		X	X	X	
Maintain STR process				X					
Evaluate/report CI Qualification Testing Process	X								
Resolve Audit Findings		X	X						

End-item Delivery Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Prepare/document version release doc				X					
Review version release doc	X				X	X	X	X	X
Approve version release doc			X						
Evaluate/report End-item Delivery Process	X								
Resolve Audit Findings		X	X						

Corrective Action Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Follow Corrective Action Process	X	X	X	X	X	X	X	X	X
Maintain Corrective Action Process				X					
Evaluate/report Corrective Action Process	X								
Resolve Audit Findings		X	X						

Certification (media certif., SW)	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Follow Certification Process	X			X			X	X	X
Certify SW	X			X					
Evaluate/report Certification Process	X								
Resolve Audit Findings		X	X						

Storage & Handling Process	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Follow Storage and Handling Process	X			X		X	X	X	X
Evaluate/report Storage and Handling Process	X								
Resolve Audit Findings		X	X						

Subcontractor Control	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Evaluate subcontractor software products	X		X	X	X	X	X	X	X
Evaluate/report Subcontractor Control Process	X								
Resolve Audit Findings		X	X						

Deviations & Waivers	**SQA Mgr**	**Prog Mgr**	**Proj Mgr**	**SCM**	**Sys Eng**	**SW Dev**	**SW Test**	**Sys Test**	**Logi stics**
Document deviations & waivers		X	X						
Recommend Approval			X						
Approve		Major	Mino r						
Evaluate/report Deviation & Waiver Process	X								
Resolve Audit Findings		X	X						

Configuration Management Process	SQA Mgr	Prog Mgr	Proj Mgr	SCM	Sys Eng	SW Dev	SW Test	Sys Test	Logi stics
Develop/Document SCM Plan				X					
Review SCM Plan	X	X	X		X	X	X	X	X
Approved SCM Plan		X	X	X					
Follow SCM processes	X	X	X	X	X	X	X	X	X
Document SCM procedures				X					
Evaluate/report CM Process	X								
Resolve Audit Findings		X	X						

SW Development Library Control Process	SQA Mgr	Prog Mgr	Proj Mgr	SCM	Sys Eng	SW Dev	SW Test	Sys Test	Logi stics
Establish SDL				X					
Follow SDL procedures	X		X	X	X	X	X	X	X
Evaluate/report SDL Process	X								
Resolve Audit Findings		X	X						

Evaluate Non-Developmental SW	SQA Mgr	Prog Mgr	Proj Mgr	SCM	Sys Eng	SW Dev	SW Test	Sys Test	Logi stics
Evaluate non-development SW	X				X	X	X	X	X
Evaluate/report Non-developmental SW Process	X								
Resolve Audit Findings		X	X						
Integrate non-development SW					X	X	X	X	X
Resolve integration errors					X	X	X	X	X

Configuration Audits	SQA Mgr	Prog Mgr	Proj Mgr	SCM	Sys Eng	SW Dev	SW Test	Sys Test	Logi stics
Assist/perform configuration audits	X			X	X	X	X	X	X
Evaluate/report Configuration Audit Process	X								
Resolve Audit Findings		X	X						

Software Quality Assurance	SQA Mgr	Prog Mgr	Proj Mgr	SCM	Sys Eng	SW Dev	SW Test	Sys Test	Logi stics
Appoint an independent SQA Auditor		X							
Assist SQA audits	X			X	X	X	X	X	X
Evaluate/report SQA Audit Process	X								
Resolve Audit Findings	X	X	X						

3.25 SCHEDULE

SQA schedules are closely coordinated with the software development schedule in reference (e). Process audits will be performed at the beginning of each new phase of development to verify that the appropriate processes are correctly implemented as defined in the planning documents. In addition, spot-checks (unscheduled audits) will be made during each phase of development to verify that the processes and desktop procedures are being followed. At the completion of a software development phase, SQA will review and report whether all steps required to transition to the next phase have been accomplished.

SECTION 4. DOCUMENTATION

The documentation that describes and supports the [[Project Name]] software or the software development process shall be created and updated periodically throughout the development cycle.

Table 4-1 is a list of [[Project Name]] software deliverable products and the associated standard or guidelines used to develop and maintain the software products. Any tailoring guidelines are also found in Table 4-1. Table 4-2 is a list of non-deliverable products.

For the project's software documents to be developed and not yet listed in Tables 4-1 and 4-2, SQA will assist in identifying the specifications, standards, and Data Item Descriptions (DIDs) to be followed in the preparation of the required documentation.

List the software products (or reference the document, e.g. CM Plan, that lists the products) that will be developed/maintained and identify the associated Data Item Description (DID) or standard or guidelines that are used to develop/ maintain the software product to which this SQA Plan applies in Table 4-1. If there are any tailoring guidelines, provide that information in Table 4-1. Identify all non-deliverable products in Table 4-2.

TABLE 4-1. DELIVERABLE SOFTWARE PRODUCTS

NOMENCLATURE	DELIVERABLE DOCUMENTATION	DID, STANDARD, GUIDELINE
[[CI Name]]	[[DOCUMENT TYPE, e.g., SSS]]	[[DID, e.g., DI-IPSC-81431 of MIL-STD-498]]
[[CI Name]]	[[DOCUMENT TYPE]]	[[DID, STANDARD, GUIDELINE]]
[[CI Name]]	[[DOCUMENT TYPE]]	[[DID, STANDARD, GUIDELINE]]

TABLE 4-2. NON-DELIVERABLE SOFTWARE PRODUCTS

DOCUMENT TITLE
[[Document title]]
[[Document title]]
[[Document title]]

State how the documents are to be checked for adequacy. The document review process should include the criteria and the identification of the review or audit by which the adequacy of each document shall be confirmed.

a. All documents will undergo a peer review in accordance with the Peer Review Process, reference (n).

Upon completion of a peer review, SQA records and reports Peer Review measurements (the item reviewed, the number of errors detected, the phase when the Peer Review was conducted, the number of closed error reports, and the number of open error reports) to SEPO.

Upon completion of a peer review, the software product will be submitted to SCM and placed under CM control. The software product will then be processed in accordance with the SCM software product approval and release process as described in reference (e).

SECTION 5. STANDARDS, PRACTICES, CONVENTIONS AND METRICS

To verify the delivery of a fully conforming, high-quality product, every individual assigned to the project will participate in quality assurance. Reference (e) defines the procedures by which the software development staff shall verify the quality of the product during the development process. The remainder of this section describes the procedures used by SQA to verify that the quality assurance provisions of this SQA Plan and applicable standards, practices, conventions, and metrics are met.

Identify the standards (mandatory requirements) to be applied. State how compliance with these items is to be monitored and assured.

[[MIL-STD-498, reference (o) or reference (b)]] is the software development standard used by the [[Project Name]] and any tailoring of this standard is documented in reference (e). Section 3 identifies SQA evaluation of the requirements, design, implementation, and test phase to verify compliance with [[references (o) or (b)]] and reference (e).

Section 4 identifies the associated DID for each software product to be developed and maintained. Any tailoring of the DID is described in reference (e). SQA will verify documentation format and content complies with the DID and reference (e).

Standards for logic structure, coding, and code comments are described in reference (e). SQA will verify source code complies with these standards as detailed in reference (e).

Standards and practices for testing are described in reference (e). SQA will verify testing activities complies with reference (e).

5.1 METRICS

Identify or reference the standards, practices, and conventions to be used in the definition, collection and utilization of software measurement data. Cite any internal (e.g., project, corporate) and external (e.g., user, customer) requirements or standards with which metrics practices must comply. IEEE Std 1045-1992, IEEE Standard for Software Productivity Metrics, reference (p) describes conventions for counting the results of the development processes. IEEE Std 1061-1992, IEEE Standard for a Software Quality Metrics Methodology, reference (q), provides a methodology for selecting and implementing process and product metrics. IEEE Std 982.1-1988, IEEE Standard Dictionary of Measures to Produce Reliable Software, reference (r) and IEEE Std 982.2-1988, IEEE Guide for using reference (r), reference (s) provide various measures for use in different life cycle phases to gain confidence in the building of reliable software. To keep metrics simple, an example of cost and schedule metrics is offered.

The following measurements will be made and used to determine the cost and schedule status of the SQA activities:

a. SQA milestone dates (planned)

b. SQA milestone dates (completed)

c. SQA work scheduled (planned)

d. SQA work completed (actual)

e. SQA effort expended (planned)

f. SQA effort expended (actual)

g. SQA funds expended (planned)

h. SQA funds expended (actual)

i. Number of noncompliance items open

j. Number of noncompliance items closed

k. Total Number of noncompliance items

SQA is responsible for reporting these measurements to the Project Manager on a monthly basis.

SECTION 6. TEST

Identify all other tests not included in verification and validation and state the methods used. Describe any testing techniques or methods that can be used to detect errors, to develop sets of test data, and to monitor computer system resources.

[[Project Name]] testing activity includes unit level testing, integration testing (at Unit and CI/HWCI level), performance testing (CI Qualification Testing), and acceptance testing (System Qualification Testing). Figure 6-1 provides the Test Process Flow. SQA shall audit the testing activities as defined in reference (e), and shall verify that the software and test documentation is subject to configuration management control. SQA shall witness the tests and verify that test results are recorded and evaluated. SQA shall coordinate the maintenance of Problem/Change Report (P/CR), sometimes called Software Trouble Report (STR), logs with SCM and shall verify that software changes are controlled according to reference (e). SQA shall witness regression testing resulting from P/CRs or STRs to verify the effectiveness of the correction.

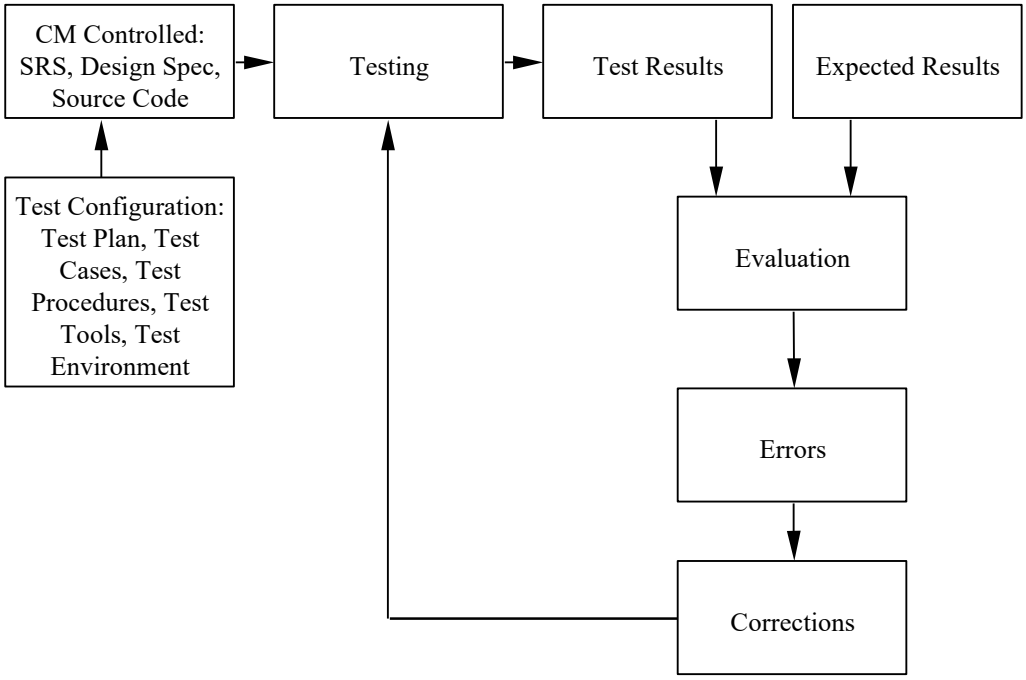

Table 6-1. Test Process Flow Diagram

This page intentionally left blank.

SECTION 7. SQA PROBLEM REPORTING AND RESOLUTION

Describe the practices and procedures to be followed for reporting, tracking, and resolving problems identified in both software items and the software development and maintenance procesess. State the specific organizational responsibilities concerned with their implementation.

This section describes the reporting and control system used by SQA to record and analyze discrepancies and to monitor the implementation of corrective action. The forms utilized by SQA for reporting are the Process Audit Report, P/CR or STR, Software Tool Evaluation Report, and Facilities Evaluation Report. Each of these forms and their uses are discussed in the following section.

7.1 PROCESS AUDIT REPORT

SQA reports the results of a process audit and provides recommendations, if necessary, using the Process Audit Report. The Process Audit Report is used to record that the process is (1) being followed correctly and working effectively, (2) being followed but is not working effectively, or (3) not being followed.

Figure 7-1 provides the format of a Process Audit Report.

7.1.1 Submittal and Disposition of Process Audit Report

The Process Audit Report is directed to the groups listed below:

a. Senior Management - The results of process audits are used in conjunction with other project status information to guide senior management attention to identify and mitigate project risks at the organizational level.

b. SEPG - The SEPG utilizes the process audits results, in conjunction with the results of audits of other projects, to identify process weaknesses and initiate or enhance process improvement in specific areas. This data becomes part of the process database so that it is available for future project analysis and use.

c. Project Manager - The project manager utilizes the report in the ways listed below:

   1. To provide insight into whether there is compliance with the development process and its effectiveness in meeting project goals. Where necessary and appropriate, the project manager may initiate enforcement activities or initiate change to the established processes using the approved procedures.

   2. To indicate agreement, disagreement, or deferral of recommendations cited in the Process Audit Report. Should the Project Manager indicate disagreement with the recommendations recorded on the Process Audit Report, the final disposition of report recommendations is made by the appropriate Project Sponsor as described in Section 7.1.2.

PROCESS AUDIT REPORT

TRACKING IDENTIFIER:_____

LEAD AUDITOR:_____ DATE OF REPORT:_____

AUDIT
TEAM:_____

_____

PROJECT NAME:_____

DATE OF AUDIT:_____

PROCESS/PROCEDURE AUDITED:_____

AUDIT CHECKLIST USED: (Attach)_____

AUDIT FINDINGS: (Check one.)

_____ Process/Procedure Acceptable

_____ Process/Procedure Conditionally Acceptable
(Subject to satisfactory completion of action items listed below)
Conditions noted:

_____ Process/Procedure Unacceptable
(Subject to satisfactory completion of action items listed below)
Conditions noted:

_____A
CTION ITEM (AI):

AI # TITLE ASSIGNED TO: DUE DATE: COMP DATE:___

_____

_____

_____

CORRECTIVE ACTION:

_____

DISPOSITION: APPROVE CANCEL DEFER

Project Manager: DATE:

_____

AI CLOSURE:

SQA Sign-off: DATE:

(FILE COMPLETED FORM IN SQA EVALUATION RECORD.)

Figure 7-1. Process Audit Report

7.1.2 Escalation Procedure for Resolution of Non-Concurrence on Process Audit Report

In the event that affected project personnel dispute the findings and recommendations of a Process Audit Report, SQA will first communicate with the affected Project Manager to resolve the dispute. If the affected Project Manager also disputes the findings and/or recommendations, the Project Sponsor (or at least one management level higher than that affected by the Process Audit Report recommended actions) directs final disposition of Process Audit Report recommendations. The affected project implements, defers, or cancels the implementation of corrective actions recommended on the Process Audit Report as directed by the Project Sponsor/Upper Level Manager. This direction is recorded and dated by the Project Sponsor (or other management, as appropriate) to be added to the SQA evaluation records of the project. SQA retains the original record of findings and subsequent resolution data in its audit files.

7.2 RECORDING PROBLEMS IN SOFTWARE CODE OR DOCUMENTATION

Problems found in the software code or documentation that is under configuration management must be recorded by means of a P/CR (or STR, as appropriate to the project) regardless of how or by whom the problem was discovered. P/CRs generated by SQA shall be prepared and processed in accordance with reference (f). SQA shall analyze P/CRs for problem trends in an effort to prevent recurring discrepancies. SQA shall report the results of P/CR trend analyses along with suggestions for problem resolution and prevention. The format of the P/CR or STR is found in reference (f).

7.3 SOFTWARE TOOL EVALUATION REPORT

Figure 7-2 provides the format for evaluating software tools as described in Section 3.2.

7.4 FACILITIES EVALUATION REPORT

Figure 7-3 provides the format for evaluating existing and planned [[Project Name]] facilities as described in Section 3.3.

SOFTWARE TOOL EVALUATION

SQA:_____ DATE OF EVALUATION:_____

Software Tool Evaluated:

Methods or criteria used in the evaluation:

Evaluation Results:

Recommended Corrective Actions

Corrective Action Taken

Figure 7-2. Software Tool Evaluation

PROJECT FACILITIES EVALUATION

SQA:_____ DATE OF EVALUATION:_____

Facility Evaluated (Equipment, User/Test/Library Space):

Methods or criteria used in the evaluation:

Evaluation Results:

Recommended Corrective Actions

Corrective Action Taken

Figure 7-3. Project Facilities Evaluation

This page intentionally left blank.

SECTION 8. TOOLS, TECHNIQUES, AND METHODOLOGIES

Identify the special software tools, techniques, and methodologies that support SQA, state their purposes, and describe their use.

Tools - SQA software tools include, but are not limited to, operating system utilities, debugging aids, documentation aids, checklists, structuring preprocessors, file comparators, structure analyzers, code analyzers, standards auditors, simulators, execution analyzers, performance monitors, statistical analysis packages, software development folder/files, software traceability matrices, test drivers, test case generators, static or dynamic test tools, and information engineering CASE tools.

Techniques - techniques include review of the use of standards, software inspections, requirements tracing, requirements and design verification, reliability measurements and assessments, and rigorous or formal logic analysis.

Methodologies - methodologies are an integrated set of the above tools and techniques. The methodologies should be well documented for accomplishing the task or activity and provide a description of the process to be used.

Where applicable, SQA will use SEPO organizational processes and tailor the processes specific to the project.

This page intentionally left blank.

SECTION 9. CODE CONTROL

The purpose of this section is to define the methods and facilities used to maintain, store, secure and document controlled versions of the identified software during all phases of the software life cycle whose appropriate use will be verified by SQA. This may be implemented in conjunction with a computer program library. This may be provided as a part of the SCM Plan. If so, an appropriate reference should be made.

Code control includes the items listed below:

a. Identifying, labeling, and cataloging the software to be controlled

b. Identifying the physical location of the software under control

c. Identifying the location, maintenance, and use of backup copies

d. Distributing copies of the code

e. Identifying the documentation that is affected by a change

f. Establishing a new version

g. Regulating user access to the code.

[[Project Name]] uses [[identify CM Code Control Software]] for code control. The code control method is described in reference (f). SQA will conduct ongoing evaluations of the code control process to verify that the process of controlling the code is effective and in compliance with reference (f). Section 3.19 further describes SQA activities for verifying the SCM process.

This page intentionally left blank.

SECTION 10. MEDIA CONTROL

The purpose of this section is to state the methods and facilities to be used, and whose proper use is to be verified by SQA, to identify the media for each computer product and the documentation required to store the media, including the copy and restore process, and to protect the computer program physical media from unauthorized access or inadvertent damage or degradation during all phases of the software life cycle. This may be provided as a part of reference (f). If so, an appropriate reference should be made.

Media control includes the items listed below:

a. Regularly scheduled backup of the media.

b. Labeled and inventoried media filed in a storage area in accordance with security requirements and in a controlled environment that prevents degradation or damage to the media.

c. Adequate protection from unauthorized access.

The software media control methods and facilities are described in reference (f). SQA will conduct ongoing evaluations of the software media control process to verify that the process of controlling the software media is effective and in compliance with reference (f). Further guidelines for SQA verification of media control are described in Section 3.16.

This page intentionally left blank.

SECTION 11. SUPPLIER CONTROL

The purpose of this section is to state the provisions by which SQA assures that software provided by suppliers meets established requirements.

Prior to any purchase of software to support the development effort, SQA and project members will define and provide complete requirements to the supplier/vendor. The Software Tool Evaluation Process will be followed. Part of the evaluation process will require the supplier or vendor to describe their technical support, handling of user questions and problems, and software product upgrades. Further guidance for SQA verification activities is described in Sections 3.17 and 3.21.

In some cases, projects do not foresee purchasing software. If that's the case, the following paragraphs may apply.

All supplier software has been operationally tested in the target system. No future supplier software is planned.

This page intentionally left blank.

SECTION 12. RECORDS COLLECTION, MAINTENANCE AND RETENTION

Identify the SQA documentation to be retained, state the methods and facilities to be used to assemble, safeguard, and maintain this documentation, and designate the retention period.

SQA activities are documented by records and reports that provide a history of product quality throughout the software life cycle. Measurement data collected will be reviewed for trends and process improvement. All SQA records will be collected and maintained in the SDL or archival storage for the life cycle of the product or a minimum of [state number of years].

This page intentionally left blank.

SECTION 13. TRAINING

Identify the training activities necessary to meet the needs of the SQA Plan.

Table 13-1 provides a matrix that identifies the required skills to perform SQA tasks to implement this [[Project Name]] SQA Plan. The training schedule will be compatible with the project schedule. In some cases, training will be conducted as On-the-Job (OJT) training.

TABLE 13-1. SQA TRAINING MATRIX

TASK	SKILL REQUIREMENTS	TYPE	SOURCE
Code Reviews	Source Language, Peer Reviews	Classroom/ OJT	SEPO, Peer Review Process and Workshop
Documentation Reviews	Software Development and Documentation standards and guidelines, Peer Reviews	Classroom/ OJT	SEPO, Peer Review Process and Workshop
Process Audits	Software Development Life Cycle Processes, Audit techniques	Classroom/ OJT	MIL-STD-498, IEEE/EIA 12207
Testing	Testing Methodologies	OJT	
SQA Management	Project Management	Classroom/ OJT	SEPO, Software Project Management (SPM) course
Metrics	Data Collection and Analysis	Classroom/ OJT	SEPO, SPM course
Problem reporting and correction action	Configuration Management	Classroom/ OJT	SEPO, CM Practitioner's Training
Tools	Vendor supplied training	Classroom/ OJT	Vendor
Code, Media, and Supplier Control	Configuration Management	Classroom/ OJT	SEPO, CM Practitioner's Training
Risk Management and Analysis	Risk Management Process	Classroom/ OJT	SEPO, SPM course
Software Management	Software Management Process	Classroom/ OJT	SEPO, Software Management for Everyone (SME) Training, SPM course

This page intentionally left blank.

SECTION 14. RISK MANAGEMENT

Specify the methods and procedures employed to identify, assess, monitor, and control areas of risk arising during the portion of the software life cycle covered by the SQA Plan.

The [[Project Name]] has developed a risk management plan as identified in [[document name]]. SQA will review and evaluate the technical risk analysis and any risk reduction plan. SQA reporting will confirm that the identified risks are managed in accordance with the provisions of the project's risk management plans, and that associated action items are reported, managed, and followed through to closure.

This page intentionally left blank.

APPENDIX A. LIST OF ACRONYMS

AI	Action Item
CDR	Critical Design Review
CMM	Capability Maturity Model
CMU	Carnegie-Mellon University
CRLCMP	Computer Resource Life Cycle Management Plan
CI	Configuration Item
DBDD	Database Design Description
DCR	Document Change Request
DID	Data Item Description
EIA	Electronic Industries Association
FCA	Functional Configuration Audit
FQR	Formal Qualification Review
HB	Handbook
HWCI	Hardware Configuration Item
IDD	Interface Design Description
IEEE	Institute of Electrical and Electronics Engineers
IRS	Interface Requirements Specification
IV&V	Independent Verification and Validation
KPA	Key Process Area
MIL	Military
NDS	Non-Developmental Software
OCD	Operational Concept Document
OJT	On-the-Job
PCA	Physical Configuration Audit
P/CR	Problem/Change Report
PDR	Preliminary Design Review
PP&O	Project Planning and Oversight
PRR	Product Readiness Review
SCM	Software Configuration Management
SDD	Software Design Document
SDF	Software Development File

SDP	Software Development Plan
SDR	System Design Review
SEI	Software Engineering Institute
SEPO	Systems Engineering Process Office
SME	Software Management for Everyone
SPAWAR	Space and Naval Warfare
SPI	Software Process Improvement
SQA	Software Quality Assurance
SRR	System Requirements Review
SRS	Software Requirements Specification
SSC	SPAWAR Systems Center
SSDD	System/Subsystem Design Description
SSR	Software Specification Review
SSS	System/Subsystem Specification
STD	Standard
STR	Software Trouble Report
SU	Software Unit
SVD	Software Version Description
TRR	Test Readiness Review
UDF	Unit Development Folder

APPENDIX B. GENERAL SOFTWARE ENGINEERING PROCESS AUDIT CHECKLISTS

PROJECT PLANNING PROCESS AUDIT CHECKLIST

Project:

Date:

Prepared by:

Procedures:

____Project Plans and commitments exist and are documented in the SDP.

____Standards governing the project's software development process are documented and reflected in the SDP.

____The content of the SDP reflects consistent implementation of organization's standard software process.

____The SDP is under configuration management.

____The activities of software estimation are conducted in accordance with Software Estimation Process and results are documented.

____The organizational database is used for making estimations.

____Software requirements are the basis for software plans, work products, and activities.

____Plans for conducting software configuration management exists and are documented in the SDP or a separate Software Configuration Management Plan (SCM Plan).

____The SCM Plan is under configuration management.

____Plans for conducting software quality assurance exists and are documented in the SDP or a separate Software Quality Assurance Plan (SQA Plan).

____Plans for conducting software integration testing exists and are documented in a Software Test Plan (STP).

____Plans for conducting system testing exist and are documented in a STP

____The STP is under configuration management.

____Plans for conducting software transition exist and are documented in a Software Transition Plan or Software Development Plan (SDP).

____Project schedules and plans undergo a peer review prior to establishing their baselines.

Figure B-1. Project Planning Process Audit Checklist

PROJECT TRACKING AND OVERSIGHT PROCESS AUDIT CHECKLIST
Project: **Date:** **Prepared by:**
Procedures: ____Project measurements are collected in accordance with the Software Measurement Plan. ____Measurements are used for re-planning analysis. ____Project Lead reviews project status on a biweekly basis. ____Branch Head reviews project status on a monthly basis. ____Division Head reviews project status on a quarterly basis. ____Quarterly Reviews are conducted in accordance with the Software Measurement Plan.

Figure B-2. Project Tracking and Oversight Process Audit Checklist

SYSTEM REQUIREMENTS ANALYSIS PROCESS AUDIT CHECKLIST

Project:

Date:

Prepared by:

Procedures:

____The correct participants are involved in the systems requirements analysis process to identify all user needs.

____Requirements are reviewed to determine if they are feasible to implement, clearly stated, and consistent.

____Changes to allocated requirements, work products, and activities are identified, reviewed, and tracked to closure.

____Project personnel involved in the system requirements analysis process are trained in the necessary procedures and standards applicable to their area of responsibility to do the job correctly.

____The commitments resulting from allocated requirements are negotiated and agreed upon by the affected groups.

____The commitments are documented, reviewed, accepted, approved and communicated.

____Allocated requirements identified as having potential problems are reviewed with the group responsible for analyzing system requirements and documents, and necessary changes are made.

____The prescribed processes for defining, documenting, and allocating requirements are followed and documented.

____A CM process is in place to control and manage the baseline.

____Requirements are documented, managed, controlled, and traced (preferably via a matrix).

____The agreed upon requirements are addressed in the SDP.

Figure B-3. System Requirements Analysis Process Audit Checklist

SYSTEM DESIGN PROCESS AUDIT CHECKLIST

Project:

Date:

Prepared by:

Procedures:

____System design documents and the traceability matrix are prepared and kept current and consistent.

____Relevant system design documents are updated based on approved requirements changes.

____Design walkthroughs (peer reviews) evaluate compliance of the design to the requirements, identify defects in the design, and alternatives are evaluated and reported.

____SQA attends a sample set of design walkthroughs. All walkthroughs are conducted.

____Defects are identified and resolved. Change control integrity is maintained.

____The content of system design documents is selectively reviewed and audited.

____Lack of compliance with standards is identified and corrective actions are determined.

____Requirements and accompanying design and tools conform to standards, and needed waivers are obtained prior to continuing software development.

____Demonstration prototypes comply with requirements and standards.

____The demonstration conforms to standards and procedures.

____The status of design milestones is reviewed.

Figure B-4. System Design Process Audit Checklist

SOFTWARE REQUIREMENTS ANALYSIS PROCESS AUDIT CHECKLIST

Project:

Date:

Prepared by:

Procedures:

Part 1. Software Requirements

____Software requirements are documented in a Software Requirements Specification (SRS) or other approved format and are included in a traceabliity matrix.

____The SRS is maintained under configuration management.

____The SRS changes undergo a peer review before they are incorporated into the requirements baseline.

____Ensure that the peer review validates testability of requirements and that appropriate metrics are established to validate measurable performance requirements.

____Software development plans, work products, and activities are changed to be consistent with changes to the software requirements.

____Software requirements analysis techniques are consistent with the SDP.

____Automated tools acquired to manage software requirements trouble reports and change requests are correctly used.

____Software engineering group is trained to perform requirements management activities.

____Measurements are made and used to determine the status of requirements management.

Part 2. Interface Requirements

____Interface requirements are documented in an Interface Requirements Specification (IRS) or other approved format.

____The IRS is maintained under configuration management.

____The IRS changes undergo peer review before they are incorporated into the requirements baseline.

____Software development plans, work products, and activities are changed to be consistent with changes to the interface requirements.

____Automated tools are used to manage interface requirements trouble reports and change requests.

____Software engineering group is trained to perform requirements management activities.

Figure B-5. Software Requirements Analysis Process Audit Checklist

SOFTWARE DESIGN PROCESS AUDIT CHECKLIST

Project:

Date:

Prepared by:

Procedures:

Part 1. Software Design

____The following documents undergo peer review during this phase of development:

- ___Software Design Document
- ___Interface Design Document
- ___Software Test Plan (Test Ids, Test Cases)
- ___Software Programmers Manual
- ___Software Test Description
- ___Firmware Support Manual

____The following modified documents are placed under CM during this phase of development:

- ___Software Design Document
- ___Interface Design Document
- ___Software Test Plan (Test Ids, Test Cases)
- ___Software Programmers Manual
- ___Software Test Description
- ___Firmware Support Manual

____Design documents and the traceability matrix are prepared and kept current and consistent based on approved software requirement changes.

____Design walkthroughs evaluate compliance of the design to the requirements, identify defects in the design, and alternatives are evaluated and reported.

____Design walkthroughs are conducted in accordance with Peer Review Process.

____Changes to the software design are identified, reviewed, and tracked to closure.

____Software design is consistent with the design methodology approved in the SDP.

Figure B-6. Software Design Process Audit Checklist

SOFTWARE DESIGN PROCESS AUDIT CHECKLIST (cont)
Project: **Date:** **Prepared by:**
Procedures: **Part 1.** Software Design (cont) ____The method, such as the Software Development Folder or Unit Development Folder, used for tracking and documenting the development/maintenance of a software unit is implemented and is kept current. **Part 2.** Interface Design ____Interface Requirements are documented in an Interface Design Document (IDD) or other approved format. ____The IDD is maintained under configuration management. ____The IDD changes undergo peer review.

Figure B-6. Software Design Process Audit Checklist (continued)

SOFTWARE IMPLEMENTATION AND UNIT TESTING PROCESS AUDIT CHECKLIST

Project:

Date:

Prepared by:

Procedures:

Part 1. Software Implementation

____Code and the traceability matrix are prepared and kept current and consistent based on approved software requirement changes.

____Code walkthroughs (peer review) evaluate compliance of the code to the approved design, identify defects in the code, and alternatives are evaluated and reported.

____Code walkthroughs are conducted in accordance with Peer Review Process.

____Changes to code are identified, reviewed, and tracked to closure.

____Code is maintained under configuration management.

____Code changes undergo peer review before they are incorporated into the software baseline.

____Software coding is consistent with the coding methodology approved in the Software Development Plan (SDP).

____The method, such as the Software Development Folder or Unit Development Folder, used for tracking and documenting the development/maintenance of a software unit is implemented and is kept current.

Part 2. Unit Testing

____Software unit testing is conducted in conformance with the approved standards and procedures described in the SDP.

____Ensure that passing criteria for unit test is documented, and that compliance has been recorded.

____Results of unit testing are documented in the Software Development Folder or Unit Development Folder.

Figure B-7. Software Implementation and Unit Testing Process Audit Checklist

UNIT INTEGRATION AND TESTING PROCESS AUDIT CHECKLIST

Project:

Date:

Prepared by:

Procedures:

Part 1. Configuration Identification

____The CIs are integrated under change control. If not, go to Part 2; otherwise, continue.

____The CIs integrated into the system were obtained from an authorized Configuration Management representative in accordance with the SDP.

____The baseline versions of each CI are integrated into the system.

____All CI components of software integration are under change control in accordance with the SCM Plan.

____If yes, describe how they are identified.

Part 2. Integration Process

____A plan for the integration of the CIs exists.

____The plan specifies the order and schedule in which the CIs are integrated.

____The CIs are integrated in accordance with the schedule and in the specified order.

____The integration plan specifies which version of each CSC and CSU is to be integrated.

____The correct version is integrated.

____The integrated CIs have completed unit testing.

____Any required corrections have been completed.

____The CIs have been retested.

____Test procedures are defined for CI integration.

____The defined procedures are followed.

Figure B-8. Unit Integration and Testing Process Audit Checklist

UNIT INTEGRATION AND TESTING PROCESS AUDIT CHECKLIST (cont)
Project: **Date:** Prepared by:
Procedures: **Part 2.** Integration Process (cont) ____Test cases are defined. ____The defined test cases are followed. ____Test pass/fail criteria are defined. ____The defined test pass/fail criteria are followed. ____The test results are documented in Unit Development Folders.

Figure B-8. Unit Integration and Testing Process Audit Checklist (continued)

CI INTEGRATION TESTING, AND SYSTEM QUALIFICATION PROCESS AUDIT CHECKLIST

Project:

Date:

Prepared by:

Procedures:

Part 1. Test Plan

____An approved test plan and test descriptions exists.

____The plan and descriptions used are under configuration management control.

____The latest version of the plan and description are used.

Part 2. Testing Process

____The system software was received from an authorized configuration management source.

____The test environment, including both hardware and software requirements, is set up as required by the test plan.

____The tests were performed in the correct order.

____Each test case in the test description was executed.

____The system is tested after each CI is integrated.

____Test passing criteria is documented and compliance is recorded.

____The results of the tests are recorded in a test report.

____If yes, describe the information that is recorded and where it is recorded.

____CIs are retested after integration to assure they still satisfy their requirements without interference from remainder of system.

____The system that results from integration of each CI is placed under configuration management control.

____If yes, describe how it is identified.

Figure B-9. CI Integration Testing and System Qualification Process Audit Checklist

CI INTEGRATION TESTING, AND SYSTEM QUALIFICATION PROCESS AUDIT CHECKLIST (cont)

Project:

Date:

Prepared by:

Part 3. Trouble Reporting

____The discrepancies found are entered into a P/CR or STR Configuration Management system for change control. If not, describe where they are they recorded.

____The entries are completed at the time the discrepancies are found.

____The P/CR or STR's reference number is kept in the test file. If not, describe where it is kept.

____P/CR or STRs are written when problems are found in the test environment, test plan, test descriptions, or test cases.

____These P/CR or STRs are sent through the same change control process as software discrepancies.

Part 4. Modifications

____Are modifications or corrections made to the test environment during testing?

____If yes, were the modifications approved through the change control process prior to implementation?

____What documentation of the modifications exists?

____What are the changes?

____Are modifications or corrections made to the test descriptions during testing?

____If yes, were the modifications approved by the change control process prior to implementation?

____What documentation of the modifications exists?

____What is the approving LCCB date? _____

____What is the STD change release date? _____

____Were the change control procedures in the SCM Plan followed?

____If no, what are the changes?

Part 5. Completion Criteria

____All specified CIs have been integrated into the system.

Figure B-9. CI Integration Testing and System Qualification Process Audit Checklist. (continued)

CI INTEGRATION TESTING, AND SYSTEM QUALIFICATION PROCESS AUDIT CHECKLIST (cont)

Project:

Date:

Prepared by:

Part 5. Completion Criteria. (cont)

____All test cases have been executed on the system.

____All P/CRs or STRs are closed out.

____If not, all outstanding P/CRs or STRs are properly documented in the VDD.

____The Software Test Report has been completed and approved.

____The Software Test Report has been placed under change control.

____The appropriate authority has determined whether the system passed or failed integration testing.

____List the individual or group that determined whether the system passed or failed.

____Describe how the pass or fail determination was made.

____The software system is ready to be integrated with operational system.

Part 6. Software Development Files

____The Software Test Report includes any retests due to software failures.

____If yes, list the failures with their corresponding P/CR or STR reference numbers.

____Using the P/CR or STR CM system, list all the CIs changed due to these failures.

____All the software development files of the listed CIs were updated in accordance with SDP.

____If not, list all software development files that were not updated.

Figure B-9. CI Integration Testing and System Qualification Process Audit Checklist. (continued)

CI INTEGRATION TESTING, AND SYSTEM QUALIFICATION PROCESS AUDIT CHECKLIST (cont)
Project: **Date:** **Prepared by:**
Part 7. Software Test Report Accuracy ____The Software Test Report supplies the Configuration Identification Number (CIN) for all test documents (STP, STD) and software. If not, the Software Test Report is incomplete. ____The tester can run the evaluation tests with the specified CINs? If not, the Software Test Report is inaccurate. ____The results of these tests match the Software Test Report? If not, the Software Test Report is inaccurate.

Figure B-9. CI Integration Testing and System Qualification Process Audit Checklist. (continued)

END-ITEM DELIVERY PROCESS AUDIT CHECKLIST

Project:

Date:

Prepared by:

Procedures:

Part 1. Audits

____A functional audit was conducted, if required.

____A physical audit was conducted, if required.

Part 2. Package Generation

____The software is generated from the software library in accordance with the SDP.

____If yes, the software is the latest version of the software in the software library.

____If not, why not?

____The documentation is generated from masters controlled by the Configuration Management personnel as required by the SCM Plan.

Part 3. Delivery Package

____The software media is labeled correctly, showing at a minimum software name, release date, and correct version number. The checklist in Figure B-12 contains more details on media labeling.

____The Version Description Document is with the software media.

____If yes, the Version Description Document is the correct version for the software.

____The Version Description Document has been formally inspected.

____The User's Manual is with the software media.

____If yes, the User's Manual is the correct version for the software.

____The User's Manual has been formally inspected.

Figure B-10. End-Item Delivery Process Audit Checklist

END-ITEM DELIVERY PROCESS AUDIT CHECKLIST (cont)

Project:

Date:

Prepared by:

Procedures:

Part 4. Media Distribution List

____A distribution list for the deliverable exists.

____If yes, it is complete, all organizations listed, all addresses correct and current.

____If yes, are any organizations listed that do not need to receive deliverable?

____Is the deliverable classified?

____If yes, the personnel on the distribution list have required clearance and need-to-know.

Part 5. Packaging

____The packaging material is suitable for the contents and transmission method used.

____Does package contain a signed transmittal letter?

____If yes, the transmittal information is correct.

____All contents are listed on transmittal contained in package.

____Does package include receipt acknowledgment form?

Part 6. Problem Notification

____There is a specified method for the receiving organization to notify the sender of problems and deficiencies in the package.

____If yes, describe the method.

____There is a specified method for logging and handling distribution problems. Describe it.

____Distribution problems are handled by a specific person. Identify that person.

Figure B-10. End-Item Delivery Process Audit Checklist (continued)

SOFTWARE CORRECTIVE ACTION PROCESS AUDIT CHECKLIST

Project:

Date:

Prepared by:

Procedures:

Part 1. Implementation of a closed loop Corrective Action (CA) Process

____The CA Process is a closed-loop.

____If yes, does the closed-loop CA Process verify the items listed below?

____All detected problems are promptly reported.

____All detected problems are entered into CA Process.

____Action is initiated on problems.

____Resolution is achieved.

____Status is tracked and reported.

____Records are maintained.

____Problem/change/discrepancy reports are the input.

____If the CA was not resolved with the Project Manager, the problem was escalated to the appropriate management personnel for resolution, and the final disposition was recorded and filed.

Part 2. Inputs to the Corrective Action Process

____A CA Process exists. Location:

____The CA Process is documented. Location:

____The CA Process is implemented.

Notes:

Figure B-11. Software Corrective Action Process Audit Checklist

SOFTWARE CORRECTIVE ACTION PROCESS AUDIT CHECKLIST (cont)

Project:

Date:

Prepared by:

Part 3. Classification of Problems by Category and Priority

____Problems are classified by category. Categories include the following.

____Software Problem. The software does not operate according to supporting documentation and the documentation is correct.

____Documentation Problem. The software does not operate according to supporting documentation but the software operation is correct.

____Design Problem. The software does not operate according to supporting documentation but a design deficiency exists.

____Problems are classified by priority. Priorities include the following.

____**Priority 1:** A software problem that does one of the following:

- Prevents the accomplishment of an operational or mission essential capability specified in the baseline requirements.
- Prevents the operator's accomplishment of an operational or mission essential capability.
- Jeopardizes personnel safety.

____**Priority 2:** A software problem that does one of the following:

- Adversely affects the accomplishment of an operational or mission essential capability specified in the baseline requirements so as to degrade performance and for which no alternative work-around solution is known.
- Adversely affects the operator's accomplishment of an operational or mission essential capability for which no alternative work-around solution is known.

____**Priority 3:** A software problem that does one of the following:

- Adversely affects the accomplishment of an operational or mission essential capability specified in the baseline requirements so as to degrade performance and for which an alternative work-around solution is known.
- Adversely affects the operator's accomplishment of an operational or mission essential capability for which an alternative solution is known.

Figure B-11. Software Corrective Action Process Audit Checklist (continued)

SOFTWARE CORRECTIVE ACTION PROCESS AUDIT CHECKLIST (cont)

Part 3. Classification of Problems by Category and Priority (cont)

____**Priority 4:** A software problem that is an operator inconvenience or annoyance and which does not affect a required operational or mission essential capability.

____**Priority 5:** All other errors.

Part 4. Performance of Trend Analysis.

____Analysis is performed to determine problem areas.

____Underlying factors/root causes are identified, categorized, and prioritized.

____Resources are expended in finding and treating root causes.

Part 5. Evaluation of Corrective Action taken

____Corrective actions are evaluated to verify the items listed below:

____Problems have been resolved.

____Adverse trends have been reversed.

____Changes have been correctly implemented.

____Introduction of no additional problems.

NOTES:

Figure B-11. Software Corrective Action Process Audit Checklist (continued)

MEDIA CERTIFICATION PROCESS AUDIT CHECKLIST

Project:

Date:

Prepared by:

Procedures:

Part 1. Media Production

____Media containing source code and media containing the object code that are delivered correspond to one another.

____A documented process that is used to implement production of media from the software library exists.

____If not, a plan is being created or a reason why one is not needed is being documented. Skip to Part 2.

____The plan was followed in production of media.

____Software was created from correct files in the software library by CM personnel.

____Documents were created from approved master copies by CM personnel.

Part 2. Media Labeling

____There is a documented standard that is followed in labeling the media.

____If yes, describe the standard method used to identify the product, version, and Configuration Identification Number.

____Media is clearly labeled.

____The label contains all required information (product, version, and Configuration Identification Number).

____If software is classified, the media clearly reflects the correct classification.

____Software document is clearly labeled with product, CIN, and version number, if applicable.

Figure B-12. Media Certification Process Audit Checklist

MEDIA CERTIFICATION PROCESS AUDIT CHECKLIST (cont)
Project: **Date:** **Prepared by:**
Part 3. Media Contents ____A listing of contents on the media exits. ____If yes, describe where the listing is located. ____The media contains the contents specified in the listing. ____The contents of the media match the label information, i.e., is it the correct version for the correct hardware platform? ____The documents contain all change pages required for this version of the documents.

Figure B-12. Media Certification Process Audit Checklist (continued)

NON-DELIVERABLE SOFTWARE CERTIFICATION PROCESS AUDIT CHECKLIST
Project: **Date:** **Prepared by:**
Procedures: ____Deliverable software is dependent on non-deliverable software. ____If yes, provision is made so acquirer has or can obtain non-deliverable software. ____Non-delivered software performs it's intended use. ____Non-delivered software is placed under configuration management.

Figure B-13. Non-Deliverable Software Certification Process Audit Checklist

STORAGE AND HANDLING PROCESS AUDIT CHECKLIST
Project: **Date:** **Prepared by:**
Procedures: ____Documents and media are stored according to the Software Development Library procedure. ____Storage areas for paper products are free from adverse environmental effects (high humidity, magnetic forces, heat, and dust) ____Storage areas for media products are free from adverse environmental effects (high humidity, magnetic forces, heat, and dust) ____Storage containers for classified material are appropriate for level of classified material.

Figure B-14. Storage and Handling Process Audit Checklist

SUBCONTRACTOR CONTROL PROCESS AUDIT CHECKLIST

Project:

Date:

Prepared by:

Procedures:

____A subcontract manager is designated to be responsible for establishing and managing the software subcontract.

____Subcontract manager is trained to perform these activities.

____The work to be subcontracted is defined and planned according to a documented procedure.

____The subcontract SOW is reviewed and approved by the project manager, branch head, and division head.

____The subcontract SOW is managed and controlled.

____The subcontractor is selected according to a documented procedure.

____The contractual agreement between the prime contractor and the software subcontractor is used as the basis for managing the subcontract. The contractual agreement documents the items listed below:

- ____The terms and conditions
- ____SOW
- ____Requirements for the products to be developed.
- ____List of dependencies between subcontractor and prime.
- ____Subcontracted products to be delivered to the prime.
- ____Conditions under which revisions to products are to be submitted.
- ____Acceptance procedures and acceptance criteria to be used in evaluating the subcontractor products before they are accepted by the prime.
- ____Procedures and evaluation criteria to be used by the prime to monitor and evaluate the subcontractor's performance.

Figure B-15. Subcontractor Control Process Audit Checklist

SUBCONTRACTOR CONTROL PROCESS AUDIT CHECKLIST (cont)

Project:

Date:

Prepared by:

Procedures:

____Subcontractor's software development plan is reviewed/approved by the prime.

____Approved subcontractor's SDP is used for tracking the software activities and communicating status.

____Changes to the subcontractor's SOW are resolved according to a documented procedure.

____Project manager conducts periodic status/coordination reviews with the subcontractor's management.

____Periodic technical reviews and interchanges are held with the subcontractor.

____Formal reviews to address the subcontractor's accomplishments and results are conducted at selected milestones.

____Reviews are documented in the SOW.

____Reviews address status of subcontractor software activities.

____Significant issues, action items, and decisions are identified and documented.

____Software risks are addressed.

____Subcontractor's SDP is refined as appropriate.

____The prime contractor's software quality assurance group monitors the subcontractor's quality assurance activities.

____The prime contractor conducts acceptance testing of subcontractor products.

____Subcontractor's performance is evaluated on a periodic basis, and reviewed with the subcontractor.

____Measurements are made and used to determine the status of the subcontract.

____The activities of the subcontract are reviewed by the Division Head on a quarterly basis.

Figure B-15. Subcontractor Control Process Audit Checklist (continued)

DEVIATION AND WAIVER PROCESS AUDIT CHECKLIST
Project: **Date:** **Prepared by:**
Procedures: ____Deviations and waivers are prepared in accordance with the documented procedures in the project SCM Plan. ____Deviations and waivers are reviewed and approved by the appropriate personnel in accordance with the project SCM Plan ____Records of deviations and waivers are maintained and reflect current development configuration.

Figure B-16. Deviations and Waivers Process Audit Checklist

SOFTWARE CONFIGURATION MANAGEMENT PROCESS AUDIT CHECKLIST

Project:

Date:

Prepared by:

Procedures:

Part 1. SCM Plan

____Project follows the organizational policy for implementing SCM.

____Group responsible for coordinating and implementing SCM for the project exists.

____A documented and approved SCM Plan is used as the basis for performing SCM activities.

____Configuration control of changes to baseline documents and software are managed in accordance with the SCM Plan.

____A configuration management library system is established as the repository for the software baseline.

____The CM library is the single place of storage for the baseline version of all software.

____Access to software products in the CM library is in accordance with the Library Control procedures.

____Software work products to be placed under SCM are identified according to the SCM Plan.

____Local Change Control Board (LCCB) exists and implements LCCB procedures.

____Change request and problem reports for all configuration items are handled in accordance with the PCR procedure.

____Changes to baselines are controlled according to the SCM Plan, LCCB procedure, and PCR procedure.

____Products from the software baseline library are created and their release is controlled according to the Library Control procedures.

____Configuration status accounting reports are prepared in accordance with the SCM plan

Figure B-17. Software Configuration Management Process Audit Checklist

SOFTWARE CONFIGURATION MANAGEMENT PROCESS AUDIT CHECKLIST (cont)

Project:

Date:

Prepared by:

Part 2. Configuration Identification

____Product Baselines and the Developmental Library can be identified.

____If yes, describe the method used to identify the Baselines and the Developmental Library.

____List the documents that make up these Baselines and Developmental Library.

____The documentation and the computer storage media containing code, documentation, or both can be identified.

____If yes, describe the method used to identify the documentation and the computer storage media and list the documents that are placed under configuration control.

____Each CI and its corresponding components can be identified.

____If yes, describe the method used to identify them.

____A method is used to identify the name, version, release, change status, and any other identification details of each deliverable item.

____If yes, describe the method used.

____For each customer, identify the deliverable item, version, release, and change status being used.

____A method is used to identify the version of each CI to which the corresponding software documentation applies.

____If yes, describe the method used.

____List the SRS and SDD for each CI.

____A method is used to identify the specific version of software contained on a deliverable medium, including all changes incorporated since its previous release.

____If yes, describe the method used.

____The deliverable medium matches the CM masters? List any discrepancies.

Figure B-17. Software Configuration Management Process Audit Checklist (Continued.)

SOFTWARE CONFIGURATION MANAGEMENT PROCESS AUDIT CHECKLIST (cont)

Project:

Date:

Prepared by:

Part 3. Configuration Control

____An established plan for performing configuration control exists.

____If yes, there is a method to establish a Developmental Library for each CI.

____If yes, describe the method used.

____List the CIs in the Developmental Library.

____A method exits to maintain current copies of the deliverable documentation and code.

____If yes, describe the method used.

____List the current copies. List all discrepancies.

____A method exits to control the preparation and dissemination of changes to the master copies of deliverable software and documentation.

____If yes, describe the method used.

____Master copies of deliverables reflect only approved changes. List any discrepancies.

____List the changes in current deliverable software/documents.

Part 4. Configuration Status Accounting

____A documented plan for implementing and performing configuration status accounting exists.

____There are status reports on all products comprising the Developmental Libraries and the Functional Allocated and Product Baselines.

____Proposed and implemented changes to a CI and its associated configuration identification documents are recorded and reported.

____If yes to two out of three, answer the following. If not, then go to Part 5.

Figure B-17. Software Configuration Management Process Audit Checklist (continued)

SOFTWARE CONFIGURATION MANAGEMENT PROCESS AUDIT CHECKLIST (cont)
Project: **Date:** **Prepared by:**
Part 4. Configuration Status Accounting (cont) ____Describe the method used to provide traceability of changes to controlled products. ____Describe the method used for communicating the status of configuration identification and associated software. ____Describe the method for ensuring that delivered documents describe and represent the associated software. **Part 5.** Engineering Change Proposals ____Engineering Change Proposals (ECPs) are prepared in accordance with the appropriate standard. ____Software Change Notices (SCNs) are prepared in accordance with the appropriate standard. ____Describe the method used for handling requested changes to the CI. ____Describe the method used to authorize SCNs and ECPs.

Figure B-17. Software Configuration Management Process Audit Checklist (continued)

SOFTWARE DEVELOPMENT LIBRARY CONTROL PROCESS AUDIT CHECKLIST

Project:

Date:

Prepared by:

Procedures:

Part 1. Establishment of SDL Environment and Control

____The SDL is established and procedures to govern its operation exist.

____The SDL provides a positive means for recognizing related elements (i.e., those versions that constitute a particular baseline and protecting the software against destruction or unauthorized modification).

____Documentation and computer program materials approved by the LCCB are placed under library control.

____All software, tools, and documentation relevant to the software development are placed under library control.

____Published procedures/standards for the SDL exist.

____SDL procedures include identification of persons/organization responsible for receiving, storing, controlling, and disseminating library materials.

____Access to SDL is limited to authorized personnel.

____If yes, describe the procedures used to limit access.

____Safeguards are in place to assure that no unauthorized alterations are made to controlled material.

____If yes, describe those safeguards.

____Describe or list the materials to be controlled.

____Describe how the materials are approved and placed under control:

____Describe how changes to the software part are handled and how changes to lines of code are identified.

____The SDL contains master copies of each CI under Computer Program Library control.

Figure B-18. Software Development Library Control Process Audit Checklist

SOFTWARE DEVELOPMENT LIBRARY CONTROL PROCESS AUDIT CHECKLIST (cont)

Project:

Date:

Prepared by:

Part 1. Establishment of SDL Environment and Control (cont)

____Periodic back up of the software is performed to prevent loss of information due to any failure of the Development Library System.

____If yes, describe the backup procedure and frequency of backups.

Part 2. Assurance of Controlled Material Validity

____Duplications from controlled and tested master copies are verified before delivery as exact copies.

____All deliverable software products that are duplicated from controlled and tested master copies are compared with that master copy to assure exact duplication.

____Describe how identification numbers and revision codes are assigned to controlled documents and software.

____Describe the way releases of controlled materials are recorded.

____List the people or organization responsible for assurance of software media validity.

____A formal release procedure exists. If so, describe it?

____The material contained in the library is promptly and correctly updated when a change to any of these materials is authorized.

Figure B-18. Software Development Library Control Process Audit Checklist (continued)

NON-DEVELOPMENTAL SOFTWARE PROCESS AUDIT CHECKLIST
Project: **Date:** **Prepared by:**
Procedures: ____Non-developmental software performs its intended functions. ____Non-developmental software is placed under internal CM. ____Data rights provisions and licensing is consistent with the SDP.

Figure B-19. Non-Developmental Software Process Audit Checklist

DOCUMENT CHANGE REQUEST (DCR)

Document Title: **[[Project Name]] Software Quality Assurance Plan**	Tracking Number:
Name of Submitting Organization:	
Organization Contact:	Phone:
Mailing Address:	
Short Title:	Date:
Change Location: (use section #, figure #, table #, etc.)	
Proposed change:	
Rational for Change:	

Note: For the [[Project Name]] to take appropriate action on a change request, please provide a clear description of the recommended change along with supporting rationale.

Send to: Commanding Officer, Space and Naval Warfare Systems Center, Code 2*XX*, 53560 Hull Street, San Diego, CA 92152-5001
Fax: *add appropriate fax number*
Email: *add appropriate email*

DCR Form 7/2003